PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 29/06, 12/26

A1

(11) International Publication Number: **WO 98/11702**

(43) International Publication Date: 19 March 1998 (19.03.98)

(21) International Application Number: PCT/US97/15837

(22) International Filing Date: 9 September 1997 (09.09.97)

(30) Priority Data:
08/712,051 10 September 1996 (10.09.96) US

(71) Applicant: ACCRUE SOFTWARE, INC. [US/US]; 83 Pioneer Way, Mountain View, CA 94041 (US).

(72) Inventors: PAGE, Robert, J.; 19963 Earls Court, Morgan Hill, CA 95037 (US). POPE, John, L.; 601 Fourth Street #204, San Francisco, CA 94107 (US). SKOLNICK, Clifford, C.; 315 Duncan Street #2, San Francisco, CA 94131 (US).

(74) Agents: GARRETT, Arthur, S. et al.; Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., 1300 I Street, N.W., Washington, DC 20005-3315 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published

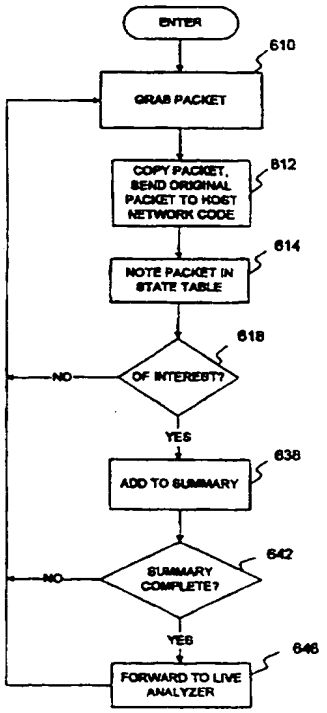
With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: APPARATUS AND METHODS FOR CAPTURING, ANALYZING AND VIEWING LIVE NETWORK INFORMATION

(57) Abstract

Apparatus and methods for viewing live information on a transmission medium receives and analyzes packets and presents the results live to a user. The received and analyzed packets may also be staged to a database for dynamic retrieval and analysis by a user. The apparatus is extensible, thus allowing flexible design, and user-definable, thus allowing systems custom tailored to a user's needs. The system is also capable of receiving data from a variety of sources other than the communications medium which is being viewed live, in order to integrate data from these other sources.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

APPARATUS AND METHODS FOR CAPTURING, ANALYZING AND VIEWING LIVE NETWORK INFORMATION

I. BACKGROUND OF THE INVENTION

The present invention relates generally to a modular data collection and analysis system, and in particular to apparatus and methods for live monitoring and analysis of packet data in a network.

As more enterprises begin to use the World Wide Web for business purposes, they need better information about the usage of their web sites or servers as well as about those accessing their web sites to improve their presentation of information and to provide better services. Unfortunately, conventional systems for monitoring and analyzing web traffic suffer some severe limitations in the data that they analyze, the type of analysis they provide, and the delays in getting the analysis.

For example, many systems for monitoring web traffic rely on log files kept in the server under examination. These log files, however, do not contain complete information because much of the important information has been stripped out before the log file is created, and because much of the information that is stored involves transactions occurring after the log was updated. Therefore, such systems, which sometimes use sophisticated statistical analysis, operate on incomplete data.

When statistics packages use such data, the results are suspect because of data loss or modification. For example, when a web server logs the time a report is sent to a client, the server usually does not know whether the client actually received the report. If the client does not receive the report, the web server cannot tell how much of the report the client received before stopping the transaction.

Other conventional systems are application-specific and can only provide detailed information about the usage of a particular application, such as a printer or some other component. These systems are limited to that application, however, and usually provide predetermined analyses.

Still other conventional systems merely monitor the physical characteristics of a network, such as the Internet. The data gathered by such systems, however, does not reflect

the content of the messages, and provides information only relevant to solving certain hardware and software problems.

Conventional systems suffer from two other limitations. One comes about because the systems do not analyze "live" information. Live information is information that is updated as it is gathered, and persons managing resources on the web need current information to make accurate adjustments. Waiting for reports or stale data renders much web management insufficient and incomplete.

Some monitoring systems provide reports in "real-time," but such "real-time" reports merely generate a static report dynamically when information is requested. The result is a report that may have been accurate when the request was made, but not after the report is received.

Finally, conventional systems that gather important data often provide that data in raw and in unanalyzed form. Doing so does not enable users of the data to appreciate the nature of the web traffic under examination.

For example, when reporting systems show every possible piece of gathered data, they are unwieldy. Even systems that reduce 200 megabyte log files into half megabyte reports still provide too much information for most users.

II. SUMMARY OF THE INVENTION

The present invention captures protocols for the packets traveling on a network, and use those protocols to extract information from the data portions of the packets. The extracted information provides a great deal of previously unavailable information about the network traffic.

In addition, the packets are analyzed "on the wire," and not from the data in the server. Thus, the full amount of data is available for analysis, not just what is placed into a log file.

Finally, by reviewing the data live and on the wire, the present invention permits live reports that are as current as possible and allow the network managers or managers of the web service to make instantaneous adjustments based on that data.

Additional advantages of the invention will be set forth in part in the description which follows, and in part will be obvious from the description, or may be learned by practice of the

invention. The advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims.

To achieve these advantages, a method of analyzing traffic on a network carrying packets of information, each packet including a data portion and a header portion with the header portion containing routing and packet identification information, comprises the steps, executed by an analyzer system, of: examining the header portions of the packets to select packets of desired types; determining protocols for the desired packets; extracting information from the data portions of the packets according to the determined protocols; and analyzing the extracted information.

A system according to this invention for analyzing traffic on a network carrying packets of information, each packet including a data portion and a header portion with the header portion containing routing and packet identification information, comprises means for examining the header portions of the packets to select packets of desired types; means for determining protocols for the desired packets; means for extracting information from the data portions of the packets according to the determined protocols; and means for analyzing the extracted information.

Both the foregoing general description and the following detailed description are exemplary and explanatory only and are not intended to restrict the claimed invention.

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate embodiments of the invention and, together with the description, explain the principles of the invention.

III. BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings:

FIG. 1 is a schematic diagram showing the flow of a communication through the layers of a network architecture;

FIG. 2 is a schematic diagram showing the format of a communication which is transferred over the communications medium;

FIG. 3 is a system diagram showing various points in a network at which the apparatus of the present system may be used;

FIG. 4 is a block diagram showing a preferred architecture in accordance with the present invention;

FIG. 5 is a block diagram showing the architecture of the Packet Processor in FIG. 4;

FIG. 6 is a flow chart demonstrating the processing performed by the Packet Processor in accordance with the present invention;

FIG. 7 is a flow chart demonstrating the processing performed by the Live Processor in FIG. 4;

FIG. 8 is a block diagram showing the architecture of the Historical Processor in FIG. 4;

FIG. 9(a) is a diagram of an example of the logical data stored in the Historical Processor in FIG. 8;

FIGS. 9(b) and 9(c) contain a block diagram of sample table in the Historical Processor in FIG. 8;

FIG. 10 is a flow chart demonstrating the Update module;

FIG. 11 is a block diagram showing the architecture of the Report Processor in FIG. 4;

FIG. 12 is a flow chart demonstrating the processing performed by the Report Processor shown in FIG. 11;

FIGS. 13(a) through 13(c) are examples of reports produced with the preferred embodiment of this invention; and

FIG. 14 is a flow chart demonstrating the processing performed by the Site Surveyor..

IV. DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A. Overview

This invention encompasses apparatus and methods for capturing and viewing live information flowing on a communications medium. The apparatus and methods may be implemented at several points in a network, as explained below. The live information captured from the communications medium is analyzed for content, and this analysis can be presented to the user as the information is being captured and analyzed.

FIG. 1 shows the layers constituting a typical communications medium, as well as the message flow through the layers. The layers shown in FIG. 1 are based on the OSI reference model.

Each end of a pathway through a communication medium uses this seven-layer protocol. The primary communication protocols are in the bottom three layers - the network layer, data link layer, and the physical layer. These layers transmit packets, frames and bits, respectively.

A transmitter develops a message starting at the highest layer, the application layer, and the system builds the rest of the message down through to the physical layer by adding information needed at the different layers. The physical layer involves the transmission of bits.

A receiver receives the bits of the message at the physical layer, and each layer strips off the portions of the communication needed by that layer until the application layer is reached.

When a user has data to transfer over the OSI architecture, he sends the data to the application layer, which places an application layer header on the data. The header and data are passed as a unit of information to the presentation layer. The presentation layer adds a presentation header to the information from the application layer.

This process of adding headers to the information from the previous layer continues through each layer. Each layer is indifferent to the content and format of the information it receives from the previous layer.

FIG. 2 is a block diagram of the message after all headers are added in the OSI architecture shown in FIG. 1. Each of the layers, starting with the addition of the application, appends its header to the message received from the previous layer. Thus, as FIG. 2 shows, the headers appear from right to left by APP (application layer), PRE (presentation layer), SES (session layer), TRA (transport layer), NET (network layer), DLK (data link layer), and PHY (physical layer). The receiver of the communication strips off the headers in the reverse order as the communication is passed up through each layer.

The Internet uses a variation of the OSI model described above, and is characterized primarily by two protocols - the Transmission Control Protocol (TCP) and the Internet

Protocol (IP). These two protocols are collectively known as "TCP/IP," or sometimes the "TCP/IP Internet Protocol Suite." This suite of protocols implements a packet-switching network.

The TCP/IP model resembles the OSI reference model in including an application layer, transport layer, and network layer (called the "internet layer").

The TCP/IP model combines the OSI data link and physical layers into a single "host-to-network" layer, but does not have the presentation and session layers. Therefore, the TCP/IP model only contains the following four layers: application, transport, internet, and host-to-network.

Examples of protocols for the Internet application layer include SMTP (electronic mail), FTP (File Transfer), DNS (Domain Name Service), TELNET (Virtual Terminal), NNTP (news articles), and HTTP (HyperText Transfer Protocol). Examples of transport layer protocols are TCP and UDP (User Datagram Protocol). IP is the primary protocol for the internet layer. The host-to-network layer may be implemented using packet radio, LAN, SATNET, or ARPANET.

The World Wide Web (WWW) is the "graphics" portion of the Internet. The feature that makes the WWW so useful for graphics is its implementation of HTTP. The HTTP handles browser requests to servers, and the server's responses to those requests.

A wealth of information is available from the various levels of the protocol. For example, such information may include the operating system, browser, version of the browser, version of the operating system, and area service provider the customer is using, as well as the time of day, geographical location, and which page visited by the customer before visiting the current page.

The present invention may be used to analyze protocol at several layers of the network architecture and carry out such analysis live. By being live on the wire, the system views all protocol levels of a message and can discern relationships among the protocol levels as messages are collected and analyzed.

The present invention uses header fields in two ways: to determine which packets are of interest and to develop network traffic information to be presented to the user. For the first

use, the analyzer system determines from its configuration which packets are of interest for the analysis it will be performing. The system then examines the headers to find those packets.

For the second use, the analyzer system uses information from the header fields to determine how to process the remaining header fields and data portions. This allows the user to specify and view information from virtually any header and data elements, or correlations of such information.

The user may either create custom reports or select predefined reports which specify how packet header and data information is to be extracted and analyzed. The analysis may include combining, or correlating, information into the visual presentation of the report. For example, a user may specify and view correlations of header portions with data portions, or header portions with other header portions.

Analysis also involves abstracting packet-level information into a form of visual presentation more easily understood by a user. For example, certain packet-level information may represent a source of the remaining information in the packet. The system according to this invention can present such information either by a text name of the source or a graphical representation of the source.

The system also correlates packet header and data information with other information. The other information can be historical data created and stored locally by the analyzer system, or the other information can be received from external entities, such as third parties and other analyzers. External information may also be either live or historical (e.g., log data). Thus, analyzers according to this invention can present live information contemporaneously with local historical information or external information from third parties or other analyzers.

External information may also define how to extract and present information to the user. For example, external information may define which fields are of interest, or how certain fields should be processed and presented to the user. Also, the reports created or specified by the user may require that external information be used to determine how to extract packet headers and analyze data information.

A. Location of Analyzer

FIG. 3 shows an analysis system in accordance with the present invention residing in different locations to analyze traffic for a server, such as server 326 on a transmission medium of a network. System 322 is placed "in-line" at a choke point for server 326. System 330 is placed on the same network 324 as server 326, essentially in parallel with the server. Additionally, a system can even be on server 326. Combinations of these locations may also used.

Each location has corresponding benefits. In the "in-line" scenario, the analysis system 322 has two network interfaces through which all traffic must pass. One interface connects to users 310 via network 314 and communications medium 318. A system manager can use this interface to account for all traffic from users 310, and then use system 322 to restrict or modify packets passing through that interface. One advantage of this placement is that the network servers do not incur any additional expense for accounting or monitoring.

In the "on-wire" scenario, analysis system 330 runs on a host on the same physical network 324 as the server 326 being monitored. This allows collection, analysis and storage without interfering with server 326.

In the "on-server" scenario, the monitoring software runs in the operating system kernel of the server 326 being monitored. This placement accounts for all the packets and can be used to restrict or modify the packets as they pass in and out of server 326. In addition, running the monitoring software on the same host reduces overall hardware costs.

B. Structure

FIG. 4 is a block diagram of the principal components of a preferred embodiment of a live monitoring and analysis system according to the invention. A user on the "Client Side" of the system interacts with the "Server Side" to create reports, select predefined reports for use, or both. The Server Side uses these reports to direct other elements of the system to develop and present live information about the network and to present historical information.

1. Server Side

The Server Side of the system connects to communications medium 412 (e.g., the Internet), and includes Network Interface 410, Packet Handler 414, Live Analyzer 418,

Historical Analyzer 426, Report Processor 430, Communication Path 428, and Site Surveyor 442. Communication Path 428 carries communications between Report Processor 430, Historical Analyzer 426, and Live Analyzer 418. In a pure software implementation of the invention, Path 428 represents the mechanisms by which the elements communicate with one another. In a hardware embodiment of the invention the path could be a bus.

Third Parties 422 include a variety of entities that exchange information with the server side. Third Parties could include information providers, databases, application programs, and other monitoring and analysis systems that collect information at another location on a network. Monitoring and analysis systems may send reports to each other to allow users at remote sites to view reports from another site. Monitoring and analysis systems may also send raw or processed data to each other, and integrate the received data with the local raw or processed data for presentation to the user in a report. Transfers of reports, raw data and processed data between monitoring and analysis systems may occur contemporaneously with ongoing packet collection and analysis, or may be staged over time.

Third Parties may also connect only to Historical Analyzer 426. This architecture may be considered more secure because all Third Party communications must then take place via a single path, the Historical Analyzer 426.

a. Network Interface 410

Network Interface 410 connects to Communications Medium 412 and collects information flowing on the medium. Network Interface 410 may be any appropriate hardware for interfacing with the network being monitored. An example of a network interface is a modem.

b. Packet Handler

FIG. 5 is a block diagram showing the basic elements of a preferred embodiment of Packet Handler 414. In the preferred embodiment, Packet Handler 414 contains Packet Grabber 514, Packet Engine 526, and State Table 522. Packet Grabber 514 connects to Network Interface 410 and Host Network Code 518. Packet Engine 526 connects to Live Analyzer 418.

Packet Handler 414 is programmed to pull packets off the network and review header information in accordance with the protocol. The header information is used to determine whether a packet is of a particular type. Once it has been determined that the particular packet is of a certain type (e.g., a web packet), the packet is analyzed further.

(1) Packet Grabber

Packet Grabber 514 is preferably a low-level software module that communicates with the Network Interface 410 and the lowest level of Host Network Code 518. In most cases, Packet Grabber 514 places the Network Interface 410 in a "promiscuous" mode to accept every packet on the network, instead of a "selective" mode in which Packet Grabber 514 accepts only packets destined for the host machine. Packet Grabber 514 makes a copy of each packet accepted and passes the copy to Packet Engine 526. Grabber 514 sends the original packet to Host Network Code 518 for normal processing.

Packet Grabber 514 preferably resides at a low level in the machine operating system. It is therefore written specifically for a specific operating system and Network Interface 410. This requires new Packet Grabbers to be created for each operating system and network interface supported. Packet Grabber 514 could also be implemented using an operating system and network layer to allow Packet Grabber 514 to be used with other operating systems and networks. Alternatively, Packet Grabber 514 could be implemented as a streams module.

(2) Packet Engine

Packet Engine 526 in the preferred embodiment is implemented as a system-independent virtual machine (VM) that executes programs for handling packets. The term "virtual machine" refers to a fictitious computer system for executing programs. The machine is "virtual" because it does not correspond to physical hardware; it only exists as a manifestation of software. Thus, a program that runs on one instance of the Packet Engine can run on any other without regard for the physical machine where it is being run. The concept of a "virtual" machine is commonly understood in the computing arts.

Packet Engine 526 inspects, modifies and transmits packets. Packet processing programs running on Packet Engine 526 instruct the Engine 526 to collect certain packets, do a preliminary analysis of them, and summarize them as "transactions." Transactions are a

sequence of packet transmissions with common characteristics. Packet Engine 526 may also accept other kinds of digital data, such as radio frequency data observed by an antenna, or sound data captured by a microphone.

Preferably, the VM emulates a machine with virtual 64-bit registers for math and relational operations. Because the concept of byte-alignment does not exist in the VM, it can perform operations on any of the bits in a bit stream without explicit shifting operations.

FIG. 6 shows the general flow of processing for Packet Handler 414. Packet Grabber 514 grabs raw packets (step 610) and feeds them to Packet Engine 526. The packet processing programs of the Packet Engine 526 determine the type of the packet by analyzing each field of the packet according to the particular protocol implementation. Packet Engine 526 may then note the packet in State Table 522 (step 614), but this step is not required if Packet Engine 526 determines no need for such an entry.

Next, Packet Engine 526 determines whether the raw packet is of interest (step 618). Packets are of interest if they contain fields having information content of interest. Packets received by Packet Engine 526 not deemed interesting are not analyzed any further, and Packet Grabber 514 retrieves the next packet (step 610). If the packet is of interest, Packet Grabber 514 adds it to the summary (step 638), and determines whether the summary is complete (step 642).

A completed transaction may be defined in a variety of ways, depending on the type of analysis undertaken. If the packet is not of interest, Packet Grabber 514 retrieves another packet (step 610). If several packets have been grouped into the summary as a complete "transaction," the completed transaction is forwarded to Live Analyzer 418 (step 646).

In the preferred implementation, the resulting summary looks like a single line with all the summary information in it. For HTTP (web traffic), the summary is roughly equivalent to the summary line that a web server writes to a web server log file. The summary that Packet Engine 526 sends to Live Analyzer 418, however, has much more information than a web server alone can collect because of the level at which the information is received from the transmission medium.

The packet processing program executed by Packet Engine 526 may also instruct Packet Engine 526 to perform a number of other actions. For example, the packet processing program may instruct the Packet Engine to send the packet to another machine or drop the packet (i.e., do not send it anywhere).

Because processing speed is an issue, the preferred embodiment has Packet Engine 526 running in the lowest levels of the operating system or kernel, whenever possible, to minimize processing overhead. In these cases, a certain amount of system-dependent code, called wrapper code, is provided as an interface into the kernel. New wrapper code must be created for each operating system to be supported.

For debugging purposes, the Packet Engine may run outside the kernel. Degraded performance, however, would recommend against placement outside the kernel for busy networks.

(3) State Table

State Table 522 is used for tracking transactions on the network. A single transaction may consist of a number of lower-level network "transmissions." Both completed and uncompleted transactions can be tracked. For example, a transaction, such as a request and response for a single web page with no embedded images, typically consists of eight network transmissions. If the web server requests that the web browser use a feature called a "cookie," the total number of packets is about sixteen in a best-case scenario. Other network factors, such as network fragmentation and retransmission and server implementation (ident request) will increase this number.

Because Packet Engine 526 reports only completed transactions, it maintains State Table 522 to keep track of transactions that have not completed. State Table 522 is preferably implemented as a simple hash table, and each transaction in the table has a relatively short life, usually only a few seconds.

Transactions complete either successfully or unsuccessfully, such as when an error condition causes the client to stop the transaction prematurely. A transaction may never complete, however, due to network problems. This happens most often when a network or host becomes unavailable for an extended period of time, for example if a terminal disconnects

from the network during a transaction. In such cases, the transaction termination is neither successful nor unsuccessful, and Packet Engine 526 keeps the uncompleted transaction in State Table 522 as long there is room. In the present embodiment, if State Table 522 grows beyond its capacity, it flushes the uncompleted transaction and sends a report to Live Analyzer 418.

State Table 522 also allows accurate collection of timing information. There are three primary times which are tracked: when a request came in, when a response was sent from the server, and when the client acknowledged completion of the page. These times allow the system to determine when the client does not receive the full page. For example, the system could determine that a user asked for a page, did not receive it after a certain period of time, and then hit the "stop" button. The system may also determine how much of the page was downloaded before the stop button was hit.

c. Live Analyzer

Live Analyzer 418, which may be written in ANSI C, acts as the real-time, live communications center of the system. It interacts with Packet Handler 414, Third Parties 422, Historical Analyzer 426, and Visualization Modules 438.

Live Analyzer 418 may be implemented as a table-driven daemon that accepts transaction summary information from Packet Handler 414, processes the information, and forwards the processed information to destinations that previously indicated an interest in receiving information by registering with Live Analyzer 418 when they start up. For instance, a Java® applet may, using a common protocol, contact Live Analyzer 418 and indicate that it is interested in receiving a live feed of specific web-related statistics.

Information providers, information collectors and other analysis systems also use Live Analyzer 418. Information providers connect to the monitoring and analysis system via the network, describe their capabilities, and start sending data. Information collectors, such as Java applets, connect and register interest in particular items. By making the protocol specification for this process available to the public, Third Parties 422 can build their own Java applets or daemons that process the data as Live Analyzer 418 generates it. Alternatively, the protocol specification may be kept proprietary and selectively released to third parties.

Live Analyzer 418 also interacts with Visualization Modules 438, described below, to provide live feed information to the user. Visualization Modules 438 function as a collection of receiver applications that present reports to the user. Other types of receiver applications, including third party applications and other analyzers, may also receive reports from the monitoring and analysis system. In this way, the user and other entities are constantly being updated with information from Live Analyzer 418.

FIG. 7 is a flow chart showing some of the processing performed by Live Analyzer 418. First, Live Analyzer 418 accepts information (step 710). Because Live Analyzer 418 gets a "full feed" from Packet Engine 526 (a complete summary of each transaction), it needs to break each summary into manageable components and analyze them before sending them to each of the components (e.g., applications, applets) that have expressed interest (step 714). Live Analyzer 418 preferably customizes the data stream for forwarding to each interested party (step 726) to minimize the amount of data needed to be transmitted to each interested application.

Data may be batched before being sent to the component (steps 718, 722). This further reduces the amount of traffic needed for monitoring. This batched stream of information is referred to as "burst live."

Preferably, each of the batched elements (data points) has a timestamp to indicate when each transaction event was generated. This allows the receiving party to "replay" the events if it chooses.

Live Analyzer 418 can send the batched summaries to short-term storage (see next section) for transfer into long-term storage at some later time, Visualization Modules 438, Java applets, other monitoring and analysis systems, and other third parties 422.

d. Historical Analyzer 426

FIG. 8 shows the components of the preferred embodiment of Historical Analyzer 426. The basic components of Historical Analyzer 426 are the Short-Term Storage 810, Update Module 814, Data Integrator 822, and Relational Database Management System (RDBMS) 826. Historical Analyzer is also connected to communicate with Third Parties 422. There are

two types of data storage in the system: Short Term Storage 810 from Live Analyzer 418, and long-term storage, such as RDBMS 826, used for historical analysis.

(1) Short-term Storage

Although the format of Short-Term Storage 810 may take a variety of forms depending upon the particular implementation, the preferred embodiment generates an ASCII file with a well-defined set of fields. Preferably, Live Analyzer 418 maintains Short-Term Storage 810, which buffers extracted data for a long enough period of time to allow transfer to the proper destination.

(2) RDBMS

Long-term storage may be implemented using a commercially available Relational Database Management System (RDBMS) 826. Tables in RDBMS 826 are generated from user-specified information. Information may in turn be presented to the user in predefined or user-defined reports.

One feature of the preferred embodiment of this invention is that RDBMS 826 only stores what is needed. For example, a user may define two reports. One contains the number of HTTP page requests per day, plus a trend graph of the number of sessions seen per day rendered as a GIF image. The other is a scatterplot of the size of HTTP responses versus the amount of time used to send each response. If these were the only reports defined on the system, only four data elements would be needed: time of HTTP request; start time of HTTP session; size of HTTP response; and time of HTTP client confirmation. This information is derived from fields of the protocol discussed above. RDBMS 826 would therefore only need a table with four elements. This "store what is needed" philosophy results in much smaller databases.

Tables for RDBMS 826 are generated according to the total number of individual data elements existing across all reports. A support program, run whenever a report is added, changed, or deleted, generates the database to produce databases of the proper size.

The preferred embodiment of the present invention uses a relational database, such as the Informix RDBMS. Other types of information storage and retrieval could be used without departing from the spirit and scope of the invention. Preferably, the information storage and

retrieval system should conveniently store, and retrieve information in a manner consistent with the user's needs. Such a system may include other types of database technology or various forms of artificial intelligence, query processing, and other database storage and retrieval schemes.

FIG. 9 (a) is a diagram of a log file listing an example of the type of data stored, and FIGS. 9 (b) and 9 (c) are examples of the data in FIG 9 (a) stored in a relational database. The log file listing of FIG. 9(a) is created by Live Analyzer 418, and is made up of application-specific and network-specific information. The particular format of the log file record may be changed depending upon system design. For example, the format may be dependent upon the particular packet processor involved. The log file is passed upstream to Update Module 814 (via Short-Term Storage 810), which converts the log file to a particular format required by the RDBMS 826.

The RDBMS format, as exemplified by FIGS. 9(a) and 9(b), is optimized for querying and reporting. Information from log files, and possibly third parties, is stored in terms of relationships among the elements of log files. The format may reflect relationships of information within log files, among log files, or over time. For example, a "hit," which is a single request from a user to a server in which the server sees the initiation of the request, may be stored in the database in a manner which establishes the relationship of the hit to other elements. In summary, RDBMS is the relational manifestation of the log files. Data Integrator 822 may also take information from Third Parties 422 and form records linked to other information in RDBMS 826.

(3) Update Module

Update Module 814 transfers information to RDBMS 826. In the preferred embodiment, Live Analyzer 418 does not feed RDBMS 826 directly. Instead, having Update Module 814 stage and filter data allows the preferred embodiment to use a slower RDBMS 826. This is advantageous because updates to RDBMS 826 usually occur much more infrequently than inputs from Live Analyzer 418.

FIG. 10 is a flow chart showing operation of Update Module 814 of Historical Analyzer 426, which may be written in ANSI C. Update Module 814 is started periodically (step 1010). Under UNIX, for example, it is started by the cron daemon.

Update Module 814 takes a file from Short-Term Storage 810 and reduces it to the data required by RDBMS 826 (step 1014), and updates RDBMS 826 (step 1018). Module 814 then returns to a waiting state.

(4) Data Integrator

Data Integrator 822 connects to Third Parties 422, Update Module 814, and RDBMS 826. Data Integrator 822 provides an external interface into RDBMS 826 to insert data into Historical Analyzer 426, and acts as a front end to hide details of the underlying database implementation.

In a preferred embodiment, Data Integrator 822 acts as a network daemon that takes information from Third Parties 422 (as well as Update Module 814), reformats the data if necessary, and inserts the new information into RDBMS 826. Integrator 822 thus permits new information to be combined with data previously collected or received from outside sources.

Data Integrator 822 also reads files written in formats commonly used by other web servers, such as files written in Common Log Format (CLF). This allows historical data developed by other systems to be integrated into the present invention.

For example, the present system may use information previously given to the web site. Accessing the web site may have required registering, which required providing user-specific information which is kept in a local data base on another web server. The other web server may have a list of their users and know the users, but not the web-usage habits of those users. The present invention system, on the other hand, knows the habits of users, but not any of the demographics. Data Integrator 822 can combine such information to permit analysis for a request. For example, a request may ask for all males under 35 that make over $30,000 a year who visit pages about surfboards.

As another example, a user may come to the site from a third-party site, which the present invention can track. This allows analysis of activities across web sites.

Data Integrator 822 may be written in ANSI C and may use a protocol provided to third parties. This allows third parties to write applications that add information for use by the analysis system in accordance disclosed herein.

e. Report Processor

FIG. 11 shows details of Report Processor 430 of FIG. 4. Report Processor 430 is the entry point for users interacting with and receiving information from the system. Report Processor 430 includes Report Generator 1110 and HTTP Daemon 1114, and is preferably implemented as a web server. The user interacts with Report Processor 430 using Web Browsers 1118. Report Generator 1110 communicates with Historical Analyzer 426 to retrieve information from RDBMS 826, and provides reports to users in the form of HTML documents, shown as Visualization Modules 438 in FIG. 4, having embedded text, graphics, and Java applets if appropriate. Report Processor 430 issues SQL requests to RDBMS 826 in accordance with the reports requested by the user.

Preferably, clients (*i.e.*, users) connect to Report Processor 430 with a standard Java-enabled web browser. If they have the correct permissions, the clients are presented with a list of reports they may view. This list is generated dynamically based on access controls set up by the webmaster.

(1) HTTP Daemon

The preferred embodiment of this invention contains an off-the-shelf HTTP Daemon 1114, but does not make any implementation-specific demands of it. Therefore, any desired HTTP daemon should be compatible with the remaining elements of the present invention as long as that daemon supports the Common Gateway Interface (CGI) and basic HTTP authentication. HTTP Daemon 1114 is not intended to operate as a main web server, only to provide a server-based interface for the analysis. HTTP Daemon 1114 does not have knowledge of the database being used nor its structures.

FIG. 12 shows the overall processing performed by HTTP Daemon 1114 and Report Generator 1110 in the preferred embodiment. When HTTP Daemon 1114 needs to send a report to a client (step 1210), it calls Report Generator 1110 to format and return a report (step 1214). This call is preferably made via the CGI, which is part of the daemon. If the requested

information has already been cached in Report Generator 1110 (step 1218), the information is taken from the cache and incorporated into the page by Report Generator 1110 (step 1226).

(2) Report Generator

If, however, the information is not already cached, Report Generator 1110, which may be implemented as a mix of ANSI C and Perl, logs into RDBMS 826, and gets meta-information (the report template) for the report (step 1220). The meta-information is created in RDBMS 826 in accordance with the information requested by the user during the report generation process, and a report is a dynamically-generated HTML page, with embedded text, images, and Java applets.

Report Generator 1110 then issues the appropriate SQL (Structured Query Language) retrieves the query results (step 1224), and incorporates the information into the page (step 1226). The HTML page is then generated (step 1230) and returned to the client (step 1236).

Building a report is computationally expensive. Therefore, Report Generator 1110 keeps the generated reports available for some period of time in case those reports are requested again. Report Generator 1110 also keeps a record of the fact that the report was requested. In this case, reports are pre-built at defined time intervals, so when clients request the report, it is fresh.

FIGS. 13 (a) - (c) show three examples of reports that Report Generator 1110 can produce. FIG. 13(a) shows a bar chart presenting a live analysis of the top ten websites that users visited before visiting the website under analysis. At the top of the bar chart is a navigation bar which may be used to select alternate reports from this page. In the example shown, the bars may fluctuate depending upon the nature of the live data.

FIG. 13(b) shows a graph of the number of hits and the number of "Resets" over time. A user will often hit a "Reset" when a website is taking too long to respond. FIG. 13 (c) shows a similar graph, but includes the number of "KeepAlives," which are indications that the user wishes to keep a communication link established for network efficiency reasons.

g. Site Surveyor

A user invokes a site survey via Request 436 to examine and assess the entities which make up the web site. In response to the user invocation, Report Processor 430 directs Site Surveyor 442 to survey the web site. Site Surveyor 442 then presents an overview to the user.

Site Surveyor is extremely helpful because composing Areas is often an arduous task. "Areas" is an important concept that will aid in appreciating the advantages of this invention. Users generally think of data in higher level terms, rather than host/domain name or URL, and the present invention allows a user to view data at a variety of levels of abstraction. For example, the present invention has defined two abstractions, as mentioned above, that provide higher-level semantics over the basic information: Groups and Areas.

A Group is a collection of users generally based on IP addresses. For instance, a "Sales" Group could indicate all the hosts in the sales organization. Similar Groups could be set up for Engineering, Marketing, and so on. Groups can consist of other groups, so the Sales Group can consist of the Groups: Milwaukee, Tampa, Baltimore, Aspen. A high-level request could be made to compare results from visitors to a website from the United States versus those from Japan or Germany. Alternatively, the comparison could analyze visitors to a website from Engineering or an analysis could break down by group the visitors to a website according to those Engineering, sales, personnel, etc. A user could then select one group, such as the sales sites. So sales sites would be clicked on, and the system provides a breakdown by sites.

An Area, on the other hand, is a collection of web pages. Typical areas may exist for Support, Press Releases, Product Information, and What's New.

Other abstractions may be defined depending on the nature of the monitoring and analysis being undertaken. These abstractions all map to, and can be derived from, information found in the network protocols discussed above.

FIG. 14 shows the general operations performed by Site Surveyor 442. It first walks the entire web site (step 1410), and analyzes the pages, the links between them, and their hierarchical structure (step 1414). Site Surveyor 442 then provides suggested Areas to the HTTP Daemon 1114 (step 1418).

The user can modify the suggested Areas using a Group and Area maintenance facility (step 1422). A web-based forms interface is used by the present invention for maintaining Groups and Areas, where they can be created, modified and deleted.

2. Client Side

The "Client Side" provides users with report creation and presentation. The user can create custom reports which are used by the system to present certain information to the user, or the user can use predefined reports. The Client Side also presents reports to the user.

Preferably, the Client Side is implemented using a web browser, as indicated generally by the broken line box. Using a web browser leverages the system design by using the extensive user interface functionality already developed for World Wide Web.

Returning to FIG. 4, the Client Side preferably includes Request 436, Report Designer 434, and one or more Visualization Modules 438. Each of these are preferably software modules which represent different capabilities performed by the present invention for the user.

a. Request

Request 436 represents any user request for service performed by Report Processor 430. This may include specific queries for data collection and analyses, site surveying, report design, or report generation.

b. Report Designer

Report Designer 434 is used to design the HTML pages of the reports. Report Designer 434 is preferably a Java applet that runs within the user's web browser and offers all the data elements defined by protocol modules defined for transferring data in the present system, as well as all the Visualization Modules that have been loaded into the system.

Report Designer 434 allows a user to design, in an interactive manner, how data and information will be displayed. Preferably, Report Designer 434 provides the user with templates and menus to design the reports. For example, Report Designer 434 allows a user to match data elements and Visualization Modules in an object-oriented fashion via a drag and drop interface.

Users select the information and visualization technique, place the resulting object on the page, and manipulate characteristics of the object, such as placement, size, text attributes,

and others. Upon finishing the page layout, the user gives the pages a name and stores the information needed to dynamically generate the page (now called a Report) on the corresponding server.

This report can be accessed like any other report, and users with appropriate permissions can view or update the report. The dynamic generation of the report is done by Report Generator 1110, described above.

The present invention may also use predefined reports, graphs, charts, tables, and other graphics tailored for specific audiences, such as a media buyer or web master or MIS Director. These can be tailored and modified by moving elements around on the screen.

c. Visualization Modules 438

One of the unique features of the present invention is the ability to view information in high-level terms without all the low-level details that conventional statistics programs provide. There are times, however, when a user would like to see more details behind the high-level summary being viewed. The preferred embodiment of this invention accomplishes this by linking one level of reports to one or more other levels. For example, on a plot of aborted page requests, a user may wish to find out more details about a particular abort. The user can then click on the plotted abort, and the system will present additional details about the selected transaction. This new report can support further exploration by providing hyperlinks and clickable areas.

In this way, Visualization Modules 438 use the familiar model of the World Wide Web to let users explore the data in a visual, interactive fashion. By using, at least in part, the preexisting structures available from the World Wide Web, the present invention leverages the use of existing technology. A customized user interface could also be developed which is not constrained to the design of World Wide Web equipment.

C. System Considerations

1. Controlling Access

Because the present invention is a Web-based system, anyone with a web browser may connect to HTTP Daemon 1114 and obtain reports. Also, using predefined protocols for applets, third-party applications and other systems similar to the present invention may obtain a live or burst live feed and request information.

Site administrators often need to restrict these kinds of accesses. They must designate who can access the system and what can be done with such access. Administrators accomplish these tasks in the preferred implementation through a web-based forms interface that allows them to add, modify and delete users and applications from lists of users and applications able to access the system. The applications have identifying characteristics, such as a log-in name and password, and the system manages restrictions placed on their access, such as the host from which the application can connect from, and the days or times of day they can connect.

2. User Access

Once in the system, users of the present invention are given permissions to do specific tasks based on how the administrator has set up the user profile. At the basic level, users may be able to view a few previously-defined reports, have access to all defined reports, or have access to the Report Designer 434 to create their own reports. Created reports can be designated as public or private.

In addition to viewing and creating or modifying reports, users may be granted additional access privileges to view or manipulate other areas of the system. For example, access privileges may be set up for Users, Applications, Groups and Areas, Modules and, System Diagnostics.

3. Application Access

Applications can use the predefined protocols for getting live (or burst-live) data from the Live Analyzer 418 and for inserting new data into RDBMS 826. The preferred protocols require an application log-in name and password, which the administrator sets up as though the application were another user. Similar to users, applications have restrictions such as where

they can connect from, what days and times they can connect, and what information they can request or send.

4. Viewing Activity

Site administrators want to see what kinds of activities have occurred on the system. For example, the site administrator may want to know who connected, and what they did when they connected. The preferred embodiment monitors user activities, logs the monitored activities, and allows the administrator to view the activity on the machine.

5. Module Maintenance

Because the preferred embodiment is implemented in modules, it can also be extended through new "modules" that can be added to the system after installation. These modules can, for example, add new collection techniques (called collection modules) and new visualization elements (visualization modules). A preferred embodiment of the system requires providers to meet predefined specifications for adding modules to the system.

Users can add new modules to the system by the Web file upload facility, as well as through the file system on the monitoring host. The preferred embodiment of the present invention contains a web forms-based user interface for managing the installed modules.

6. System Diagnostics

The preferred embodiment of the present invention has a web-based interface for viewing the hardware, operating system, network connections, software revisions, and other pieces of data. This interface lets the administrator know how the monitoring system is behaving, and can relay the information to service personnel if necessary.

7. Samples Of Web-Related Data

There are a number of algorithms to transform the raw "on-the-wire" data into higher-level information. Some few examples of transforms include: 1) converting "hits" to "sessions"; 2) determining a user's Internet connection speed; 3) determining the total time it takes to load a "complete" page (where "complete" means HTML + images + applets, etc.); and 4) determining how often users hit their browser Stop button, and at what point in the page (time elapsed, % transferred).

One advantage of the present invention is that, unlike systems that use what are essentially statistics packages, collected data is highly accurate. A second advantage is that the analysis provided by the present system is targeted and useful, not merely tables and raw numbers. Finally, the system is open and extensible, thus allowing the basic architecture to grow and change as the software is improved upon, or as different user needs arise.

8. On-the-wire data collection

The present invention achieves these advantages through several features. One is that the invention requires data to be collected on the transmission medium. To collect data on the transmission medium, the embodiments of the invention first capture the protocol by stripping down each layer of the protocol and using the protocol to analyze the contents and the transmission.

Also, the present invention lives on the wire, so it sees these transactions and reports on them live. The invention can thus present to users data showing exactly what is happening on the transmission medium as it is happening. By updating the data live, the invention allows the user to have current data without having to remember to request updates.

Another feature of the present invention is the extensive support for storage of whatever data is needed by the user. Live information does not provide a picture of the communications medium because current data does not show trends, allow planning for upgrades, or show results of changes. Proper storage of historical data, such as in a relational database, permits fast retrieval of data when necessary for such tasks.

a. Third Party Information

Collecting data is not always enough, however, because all the data needed for a report may not be readily available from the network. For example, while the present invention can provide users with views of live network activity, it can not provide views of account records for the people using the network. Open interfaces into and out of the system for Third Party access allows users can integrate data from any other source or sources and analyze the entire data set as one.

For example, the user can begin with high-level summaries through intelligent report design. If they choose to, the users can then go beyond the summaries into the raw numbers.

Users may also view information which helps them understand how the numbers relate to others. The present invention provides an interactive "data exploration" model in which a user navigates through the information instead of being overwhelmed by it. This "data exploration" feature leverages the World Wide Web, so there's no retraining or new interfaces to learn.

In addition, making both live and historical data available provides a mechanism for adding new information from Third Parties to the relational database. An Application Programs Interface (API) is used for programming new Java applets. The Java applets provide interactive visualization and assists the user in receiving new information.

Although the system is designed for efficiency, it can also be ported to other platforms because it is designed modularly and layered to provide easy integration among the various layers. Making the system modular and layered isolates the machine-dependent parts from the rest of the components. At the user interface, for example, everything is constructed so it runs on many platforms without porting.

The system-level software may advantageously be written using the C language. The application-level software may advantageously be written in C and Perl, and the interactive visualization software may be written in Java.

D. Other Considerations

1. Encryption

One possible limitation arises when those sending messages use encryption to prevent eavesdropping. Because the analysis system according to the present invention relies on access to the unencrypted traffic to perform analysis, encryption prevents the systems from getting detailed information about the transaction from the network.

Encryption does not totally frustrate use of this invention, however, because even encrypted messages, however, the analysis system can get basic transaction information, such as the fact that a request was made, a response was sent and acknowledged (or aborted), when these events occurred, and the parties involved.

Because the present invention can also get transaction information from other sources, a system monitoring and analyzing a web server can integrate the collected data with a log file generated by the web server. The correlation of these two data sets can result in powerful

analysis unavailable from regular log file analysis. For example, all the transaction timing information is still available, as well as download times, and the percentage of traffic that is or is not encrypted.

Furthermore, the data may be decrypted at the firewall or another access point on virtual private networks. Doing so provides the full benefits of the invention.

2. SNMP

The present invention is not a full-fledged network management system. Rather, it is designed specifically for high-level, live and historical network analysis. Thus, it does not have its own SNMP MIB (Simple Network Management Protocol Management Information Base, an Internet standard for managing network components) or management interface, other than what HTML and the published protocols provide. For administrators wishing to be notified when certain thresholds are crossed, however, a custom proxy agent could be developed to request the information from the system and feed it into the appropriate management system. Also, because web servers may support the emerging Web Server MIB, the preferred embodiment of the present invention also has a component that collects information from the web server using SNMP and integrates it into the database.

V. CONCLUSION

The present invention obtains the advantages of live data reporting and providing tools for accurate and in-depth analysis of data and network traffic by capturing data on the wire and looking at the data portions of messages. This invention also allows reports that efficiently and accurately summarize and communicate analyses.

Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the disclosed embodiments. The specification and examples are exemplary only, and the true scope and spirit of the invention is defined by the following claims and their equivalents.

WHAT IS CLAIMED IS:

1. A method of analyzing traffic on a network carrying packets of information, each packet including a data portion and a header portion with the header portion containing routing and packet identification information, and the method comprising the steps, executed by an analyzer system, of:

examining the header portions of the packets to select packets of desired types;

determining protocols for the desired packets;

extracting information from the data portions of the packets according to the determined protocols; and

analyzing the extracted information.

2. The method of claim 1 further including the step of

generating a report based on the analyzed extracted information showing selected characteristics of network traffic.

3. The method of claim 2 wherein the step of generating a report further includes the substep of

updating the report as information is extracted to provide live reporting of the network traffic on the network.

4. The method of claim 2 wherein the substep of generating a report further includes the substep of

sending the report to a receiver application on the network.

5. The method of claim 2 wherein the substep of generating a report further includes the substep of

sending the report to another analyzer system.

6. The method of claim 1 wherein the step of analyzing the extracted information includes the substep of

analyzing the extracted information as the information is extracted to provide live reporting of network traffic.

7. The method of claim 1 wherein the network is the world wide web, and wherein the step of examining the header portions includes the substep of

selecting web packets.

8. A method of analyzing traffic on a network carrying packets of information, each packet including a data portion and a header portion with the header portion containing routing and packet identification information, the method comprising the steps, executed by an analyzer system, of:

examining the header portions of the packets to select packets of desired types;

determining protocols for the desired packets;

extracting information from the data portions of the packets according to the determined protocols;

receiving auxiliary information; and

analyzing the extracted information and the auxiliary information.

9. The method of claim 8 wherein the step of receiving auxiliary information includes the substep of

extracting the auxiliary information from the header portions of the corresponding packets.

10. The method of claim 8 wherein the step of receiving auxiliary information includes the substep of

receiving the auxiliary information from a source external to the analyzer system.

11. The method of claim 10 wherein the step of receiving auxiliary information from an external source includes the substep of

receiving the auxiliary information from an external database system.

12. A method of measuring the activity of a network participant on a network carrying packets of information, each packet including a data portion and a header portion, the header portion containing routing and packet identification information, and the method comprising the steps, executed by an analyzer system, of:

intercepting packets on the network involving the network participant;

examining the header portions of the packets to select packets of desired types;

determining protocols for the desired packets;

extracting information regarding the network participant from the data portions of the packets according to the determined protocols; and

analyzing the extracted information.

13. The method of claim 12, wherein the network participant is a user, and

wherein the step of extracting information includes the substep of

extracting information about messages sent by that user to a selected site on the network.

14. The method of claim 12, wherein the network participant is a server, and

wherein the step of extracting information includes the substep of

extracting information about messages sent to or from the server.

15. The method of claim 12, further including the step of

receiving auxiliary information; and

wherein the step of analyzing the extracted information includes the step of

analyzing the extracted information with the auxiliary information.

16. The method of claim 12 further including the step of

generating a report based on the analyzed extracted information showing selected characteristics of network traffic.

17. The method of claim 16 wherein the step of generating a report further includes the substep of

updating the report as information is extracted to provide live reporting of the network traffic on the network.

18. The method of claim 16 wherein the substep of generating a report further includes the substep of

sending the report to a receiver application on the network.

19. The method of claim 16 wherein the substep of generating a report further includes the substep of

sending the report to another analyzer system.

20. The method of claim 12 wherein the step of analyzing the extracted information includes the substep of

analyzing the extracted information as the information is extracted to provide live reporting of network traffic.

21. A method of measuring the activity of a network participant on a network carrying packets of information, each packet including a data portion and a header portion, the header portion containing routing and packet identification information, and the method comprising the steps, executed by an analyzer system, of:

intercepting packets on the network involving the network participant;

examining the header portions of the packets to select packets of desired types;

determining protocols for the desired packets;

extracting information regarding the network participant from the data portions of the packets according to the determined protocols;

collecting usage information about of an application on a processor on the network; and

analyzing the extracted information and the usage information.

22. The method of claim 21, wherein the network participant is a user, and

wherein the step of extracting information includes the substep of

extracting information about messages sent by that user to a selected site on the network.

23. The method of claim 21, wherein the network participant is a server, and

wherein the step of extracting information includes the substep of

extracting information about messages sent to or from the server.

24 The method of claim 21, further including the step of

receiving auxiliary information; and

wherein the step of analyzing the extracted information includes the step of

analyzing the extracted information with the auxiliary information.

25. The method of claim 21 further including the step of

generating a report based on the analyzed extracted information showing selected characteristics of network traffic.

26. The method of claim 25 wherein the step of generating a report further includes the substep of

updating the report as information is extracted to provide live reporting of the netowrk traffic on the network.

27. The method of claim 25 wherein the substep of generating a report further includes the substep of

sending the report to a receiver application on the network.

28. The method of claim 25 wherein the substep of generating a report further includes the substep of

sending the report to another analyzer system.

29. The method of claim 21 wherein the step of analyzing the extracted information includes the substep of

analyzing the extracted information as the information is extracted to provide live reporting of network traffic.

30. A method, executed by an analyzer system, of analyzing world wide web packets having a header portion and a data portion, comprising the steps of:

examining header portions of the packets;

selecting packets of desired types based on the header portions;

extracting information from the selected packets based on information from header portions; and

analyzing the extracted information to report on world wide web packet traffic.

31. The method of claim 30, wherein the step of analyzing the extracted information includes the substep of

analyzing the extracted information as it is extracted.

32. The method of claim 30, wherein the analyzing step includes the substep of:

combining the extracted information with previously extracted information.

33. The method of claim 30, wherein the analyzing step includes the substep of:

combining the extracted information with third-party information.

34. The method of claim 30 further including the step of

generating a report based on the analyzed extracted information showing selected characteristics of network traffic.

35. The method of claim 34 wherein the step of generating a report further includes the substep of

updating the report as information is extracted to provide live reporting of the network traffic on the network.

36. The method of claim 34 wherein the substep of generating a report further includes the substep of

sending the report to a receiver application on the network.

37. The method of claim 34 wherein the substep of generating a report further includes the substep of

sending the report to another analyzer system.

38. The method of claim 34 wherein the step of analyzing the extracted information includes the substep of

analyzing the extracted information as the information is extracted to provide live reporting of network traffic; and

wherein the substep of generating a report further includes the substep of

updating the report live.

39. The method of claim 34 wherein the substep of generating a report further includes the substep of

receiving inputs from a user to design the report.

40. A system for analyzing traffic on a network carrying packets of information, each packet including a data portion and a header portion with the header portion containing routing and packet identification information, the system comprising:

means for examining the header portions of the packets to select packets of desired types;

means for determining protocols for the desired packets;

means for extracting information from the data portions of the packets according to the determined protocols; and

means for analyzing the extracted information.

41. The system of claim 40 wherein the traffic analyzed is the traffic related to a selected server on the network; and

wherein the system resides on a communications path coupled to the selected server.

42. The system of claim 41 wherein the communications path carries all network traffic into and out of the selected server.

43. The system of claim 41 wherein the communications path is in parallel with the selected server.

44. The system of claim 40 wherein the traffic analyzed relates to a selected server on the network; and

wherein the system resides in the selected server.

45. The system of claim 40, wherein the means for analyzing the extracted information includes

means for temporarily storing the extracted information.

46. The system of claim 45, wherein the means for analyzing the extracted information includes

a database, coupled to the means for temporarily storing, to provide longer term storage of the extracted information.

47. The system of claim 46, wherein the means for for analyzing the extracted information includes

means, coupled to the database, for storing auxiliary information with the extracted information.

48. The system of claim 40, further including

means, coupled to the analyzing means, for generating a report including the analyzed information.

49. The system of claim 48, wherein the means for generating a report includes means for updating the report as information is extracted to provide a live report.

50. A network carrying packets of information, each packet including a data portion and a header portion, the header portion containing routing and packet identification information and the network comprising:

a server providing network functions; and

a network analyzer system including

means for examining the header portions of the packets to select packets of desired types;

means for determining protocols for the desired packets;

means for extracting information from the data portions of the packets according to the determined protocols; and

means for analyzing the extracted information to report on the traffic on the network.

51. The network of claim 50, wherein the network analyzer system resides on a communications path coupled to the selected server.

52. The system of claim 51 wherein the communications path carries all network traffic into and out of the server.

53. The system of claim 52 wherein the communications is in parallel with the selected server.

54. The system of claim 51 wherein the network analyzer system resides in the selected server.

55. The system of claim 50, wherein the means for analyzing the extracted information includes

means for temporarily storing the extracted information.

56. The system of claim 55, wherein the means for for analyzing the extracted information includes

a database, coupled to the means for temporarily storing, to provide longer term storage of the extracted information.

57. The system of claim 56, wherein the means for for analyzing the extracted information includes

means, coupled to the database, for storing auxiliary information with the extracted information.

58. The system of claim 50 further including

means, coupled to the analyzing means, for generating a report including the analyzed information.

59. The system of claim 58, wherein the means for generating a report includes

means for updating the report as information is extracted to provide a live report.

60. A system for analyzing traffic on a network carrying packets of information, each packet including a data portion and a header portion with the header portion containing routing and packet identification information, the system including

a packet handler for receiving the packets from a network interface and examining the header portions of the packets to select packets of desired types; and

an analyzer engine, coupled to the packet handler, for extracting and segregating information from the data portions of the packets according to predetermined protocols.

61. The system of claim 60 further including

a report processor, coupled to the analyzer engine, for generating predetermined reports from the extracted information.

62. The system of claim 60 wherein the analyzer engine includes

a historical processor for storing previously extracted information.

63. The system of claim 60 wherein the packet handler includes

a packet grabber to copy of each packet received from the network interface.

64. The system of claim 63 wherein the packet grabber includes

means for sending the original packet to the network interface.

65. The system of claim 60 wherein the packet handler includes

a packet engine to determine the type of the packet.

66. The system of claim 65, wherein the packet engine includes

means for determining whether to select the packet based on the packet type.

67. The system of claim 65, wherein the packet engine includes

means for determining whether to send the packet outside the system.

68. The system of claim 60 wherein the packet handler includes a state table to track the status of transactions on the network.

69. The system of claim 69, wherein the packets are part of transactions, and wherein the state table includes means for keeping track of transactions that have not completed.

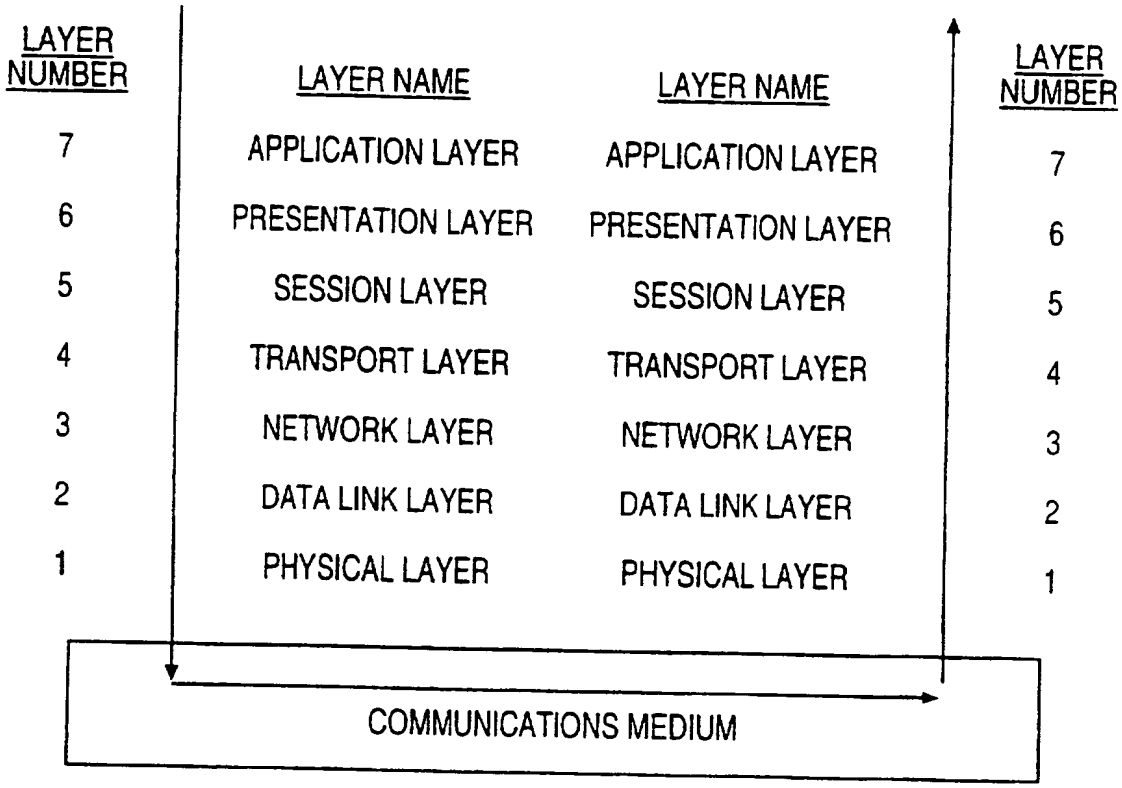

LAYER NUMBER
LAYER NAME
LAYER NAME
LAYER NUMBER
7
APPLICATION LAYER
APPLICATION LAYER
7
6
PRESENTATION LAYER
PRESENTATION LAYER
6
5
SESSION LAYER
SESSION LAYER
5
4
TRANSPORT LAYER
TRANSPORT LAYER
4
3
NETWORK LAYER
NETWORK LAYER
3
2
DATA LINK LAYER
DATA LINK LAYER
2
1
PHYSICAL LAYER
PHYSICAL LAYER
1
COMMUNICATIONS MEDIUM

FIG. 1


PHY
DLK
NET
TRA
SES
PRE
APP
DATA

FIG. 2

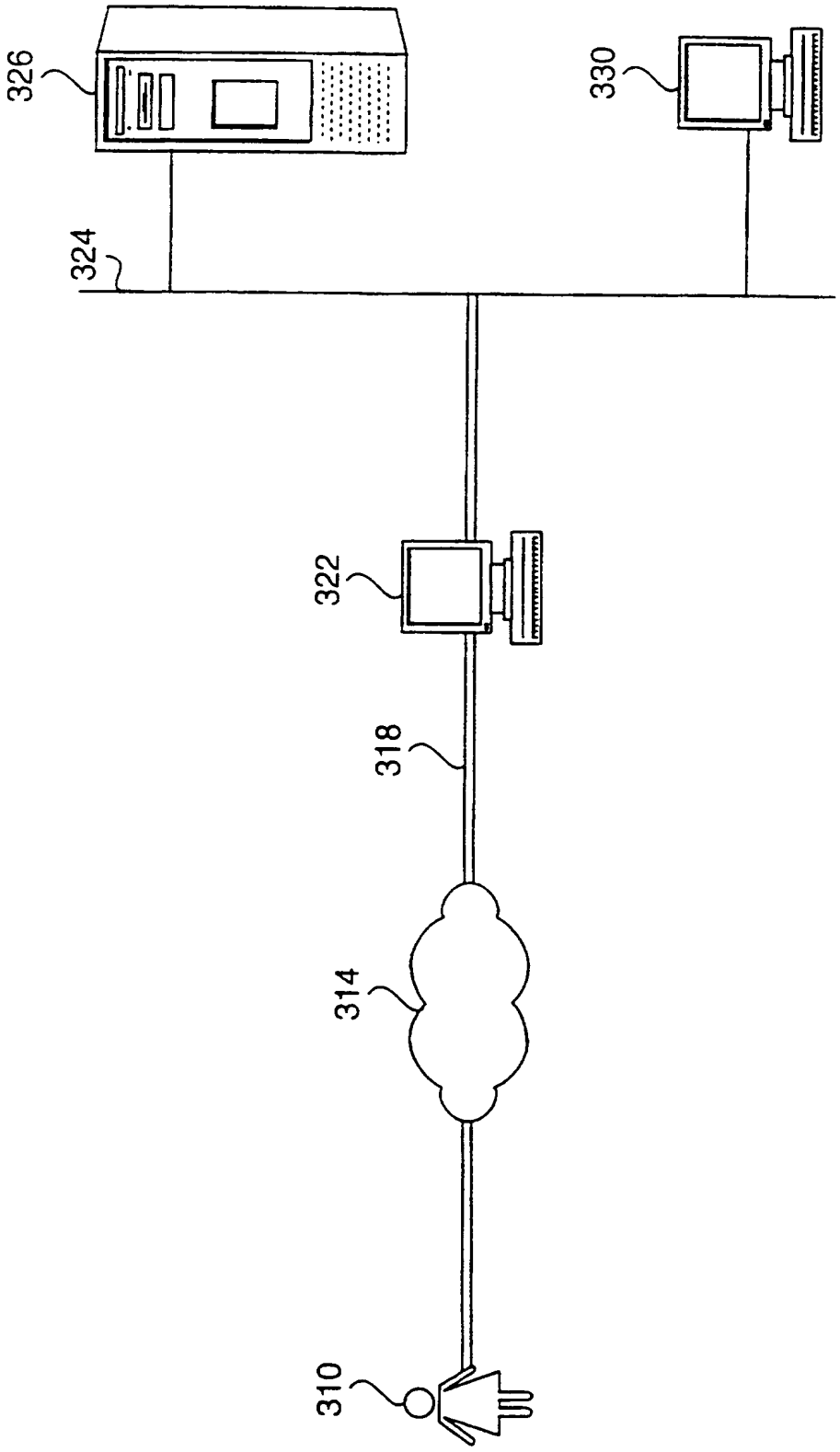


FIG. 3

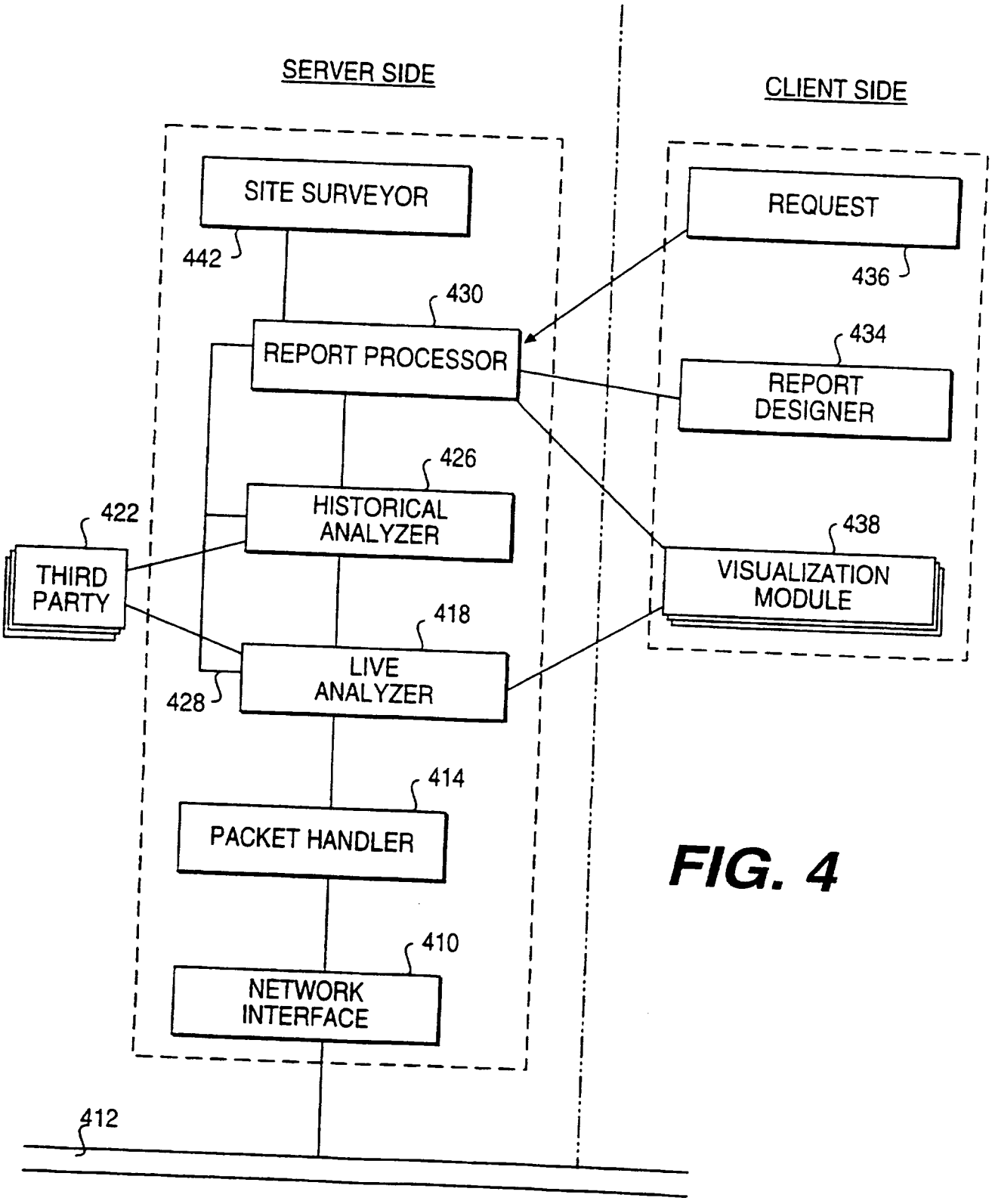

SERVER SIDE
CLIENT SIDE
SITE SURVEYOR
442
REQUEST
436
430
REPORT PROCESSOR
434
REPORT DESIGNER
426
HISTORICAL ANALYZER
422
THIRD PARTY
438
VISUALIZATION MODULE
418
LIVE ANALYZER
428
414
PACKET HANDLER
410
NETWORK INTERFACE
412

FIG. 4

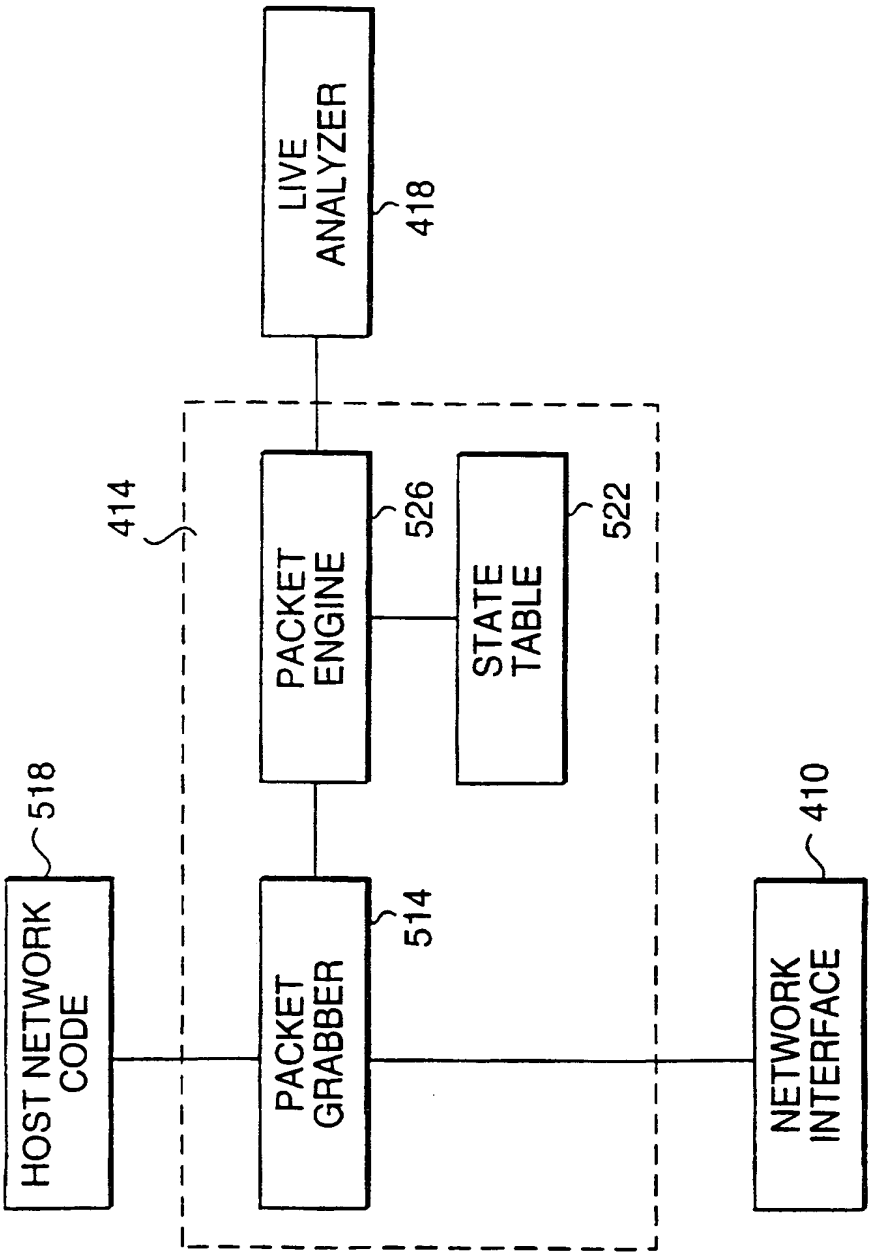


FIG. 5

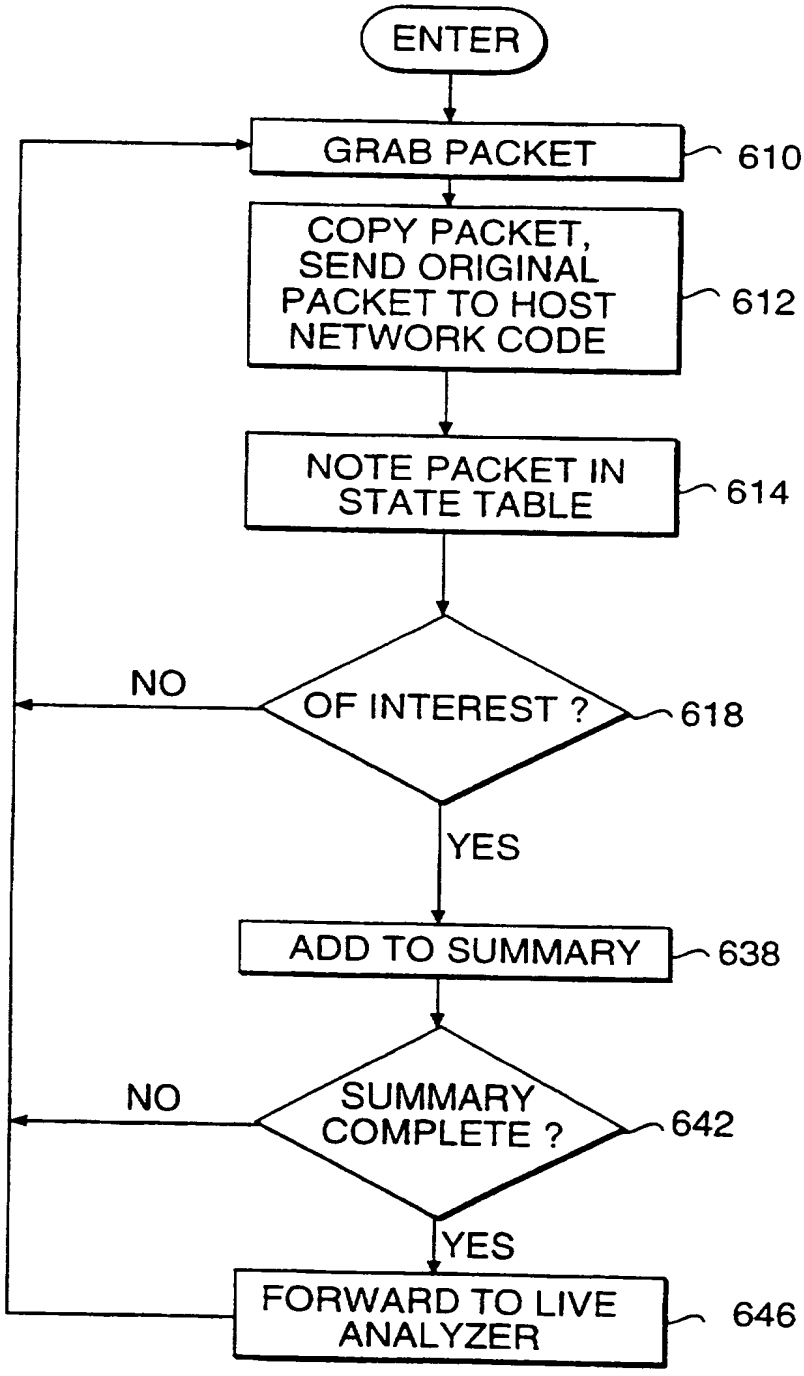

ENTER
GRAB PACKET
610
COPY PACKET, SEND ORIGINAL PACKET TO HOST NETWORK CODE
612
NOTE PACKET IN STATE TABLE
614
NO
OF INTEREST ?
618
YES
ADD TO SUMMARY
638
NO
SUMMARY COMPLETE ?
642
YES
FORWARD TO LIVE ANALYZER
646

FIG. 6

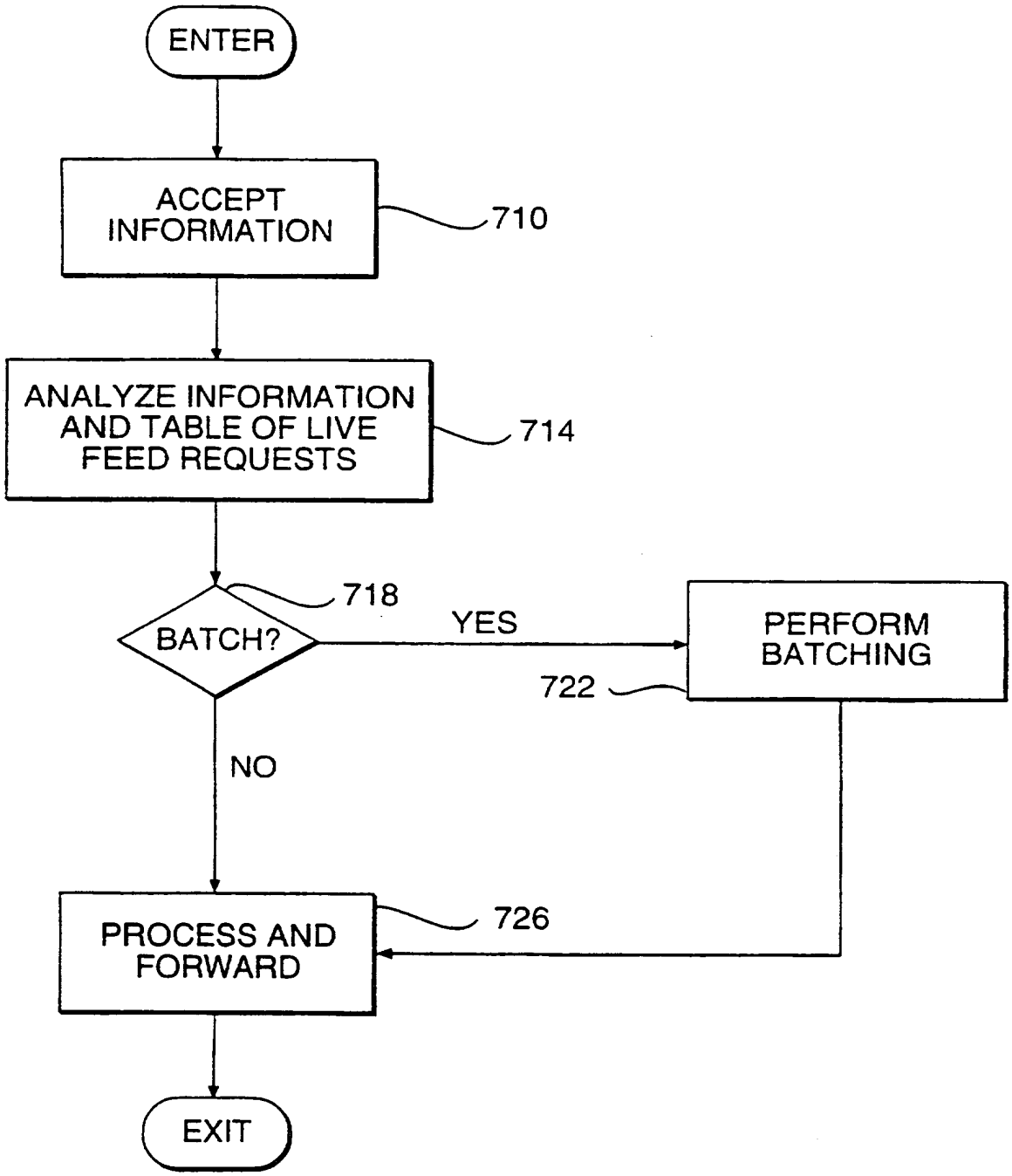

ENTER
ACCEPT INFORMATION
710
ANALYZE INFORMATION AND TABLE OF LIVE FEED REQUESTS
714
718
BATCH?
YES
PERFORM BATCHING
722
NO
PROCESS AND FORWARD
726
EXIT

FIG. 7

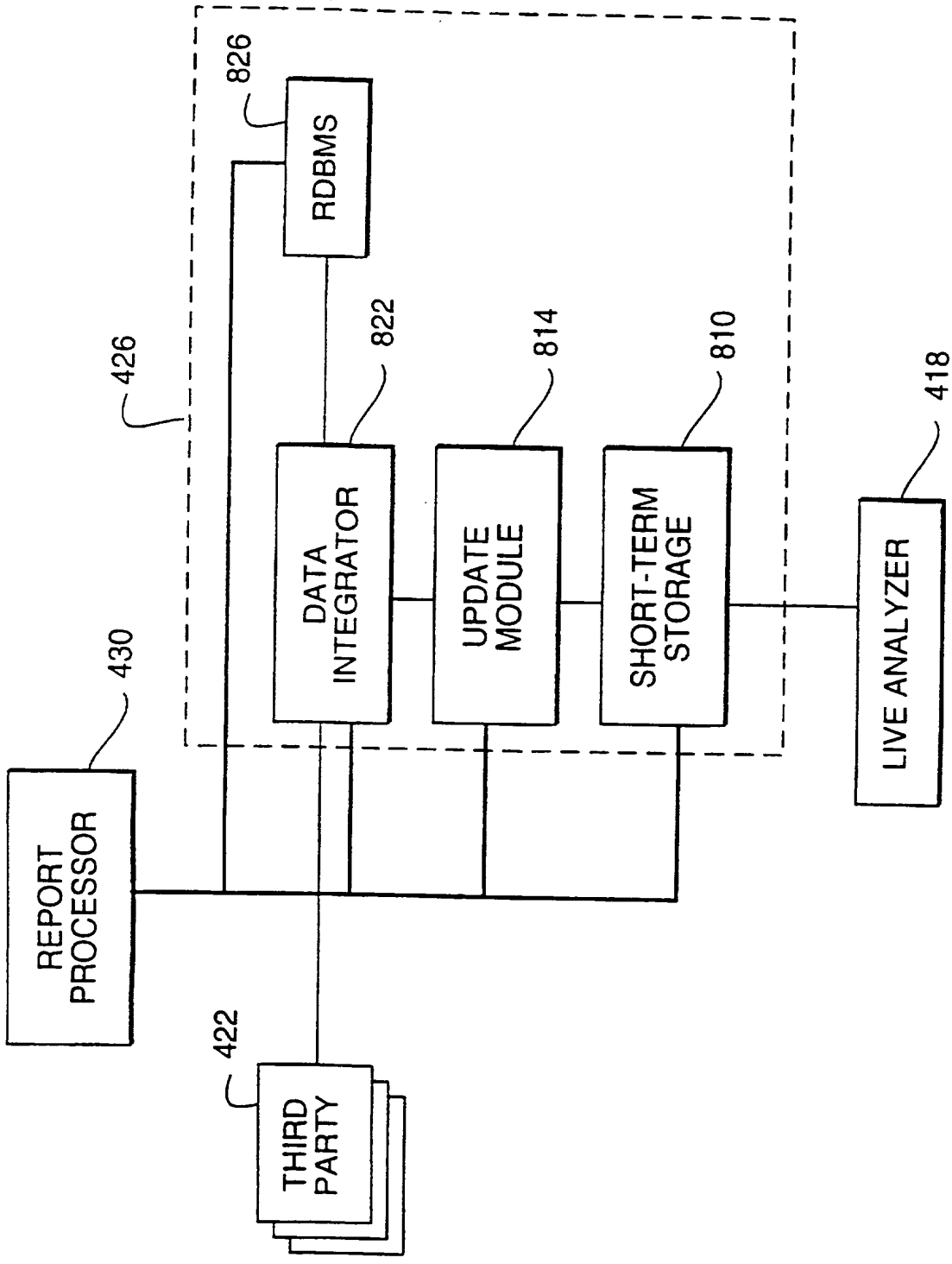

826
RDBMS
822
814
810
418
426
DATA INTEGRATOR
UPDATE MODULE
SHORT-TERM STORAGE
LIVE ANALYZER
430
REPORT PROCESSOR
422
THIRD PARTY

FIG. 8

```
# TIMING:
#         - LOGFILE TIMESTAMP (TSTAMP)
#         - INITIAL TIMESTAMP (T0)
#         - ROUND-TRIP TIME (T2 - T1)
#         - SERVER RESPONSE TIME (T3 - T2)
#         - CONNECTION TIME (T4 - T0)
#
#======================
# NAME              MEANING
#======================
IP_SRC              IP SRC ADDRESS (CLIENT)
IP_DST              IP DST ADDRESS (SERVER)
REQ_METHOD          HTTP REQUEST METHOD (E.G. GET OR POST)
REQ_BUF             1ST LINE OF HTTP REQUEST
CTYPE               MIME CONTENT-TYPE
RSPCODE             HTTP RESPONSE CODE
TSTAMP              TIMESTAMP FOR START OF CONNECTION
T0                  (T0 = INITIAL CLIENT SYN)
T1                  (T1 = DELTA: INITIAL SERVER SYN)
T2                  (T2 = DELTA: CLIENT ACK TO SERVER SYN)
T3                  (T3 = DELTA: CLIENT GET OR POST)
T4                  (T4 = DELTA: SERVER RESPONSE CONTAINING DATA)
T5                  (T5 = DELTA: CLOSE OF CONNECTION)
AUTH                HTTP AUTHENTICATION INFO
NBYTES              # OF BYTES REPORTED SENT
SBYTES              # OF BYTES ACTUALLY SENT
SETCOOKIE           HTTP "SET COOKIE" VALUE
COOKIE              HTTP "COOKIE" VALUE
REFERRER            HTTP "REFERER" VALUE
USERAGENT           HTTP "USER-AGENT" VALUE
SCRPORT             TCP SOURCE PORT (CLIENT)
DSTPORT             TCP DESTINATION PORT (SERVER)
STATUS              TCP CONNECTION INFO:
                              R = CLIENT RESET
                              X = CLIENT RETRANSMISSION
                              Y = RST FROM SERVER
                         (DEBUGGING INFO:)
                              I = INITIAL CLIENT SYN RECEIVED
                              S = FIRST SERVER SYN SENT
                              H = 3-WAY HANDSHAKE COMPLETE
                              K = HTTP KEEP ALIVE USED
                              D = DATA SENT BY SERVER
                              Z = FIN FROM SERVER
```

FIG. 9(A)

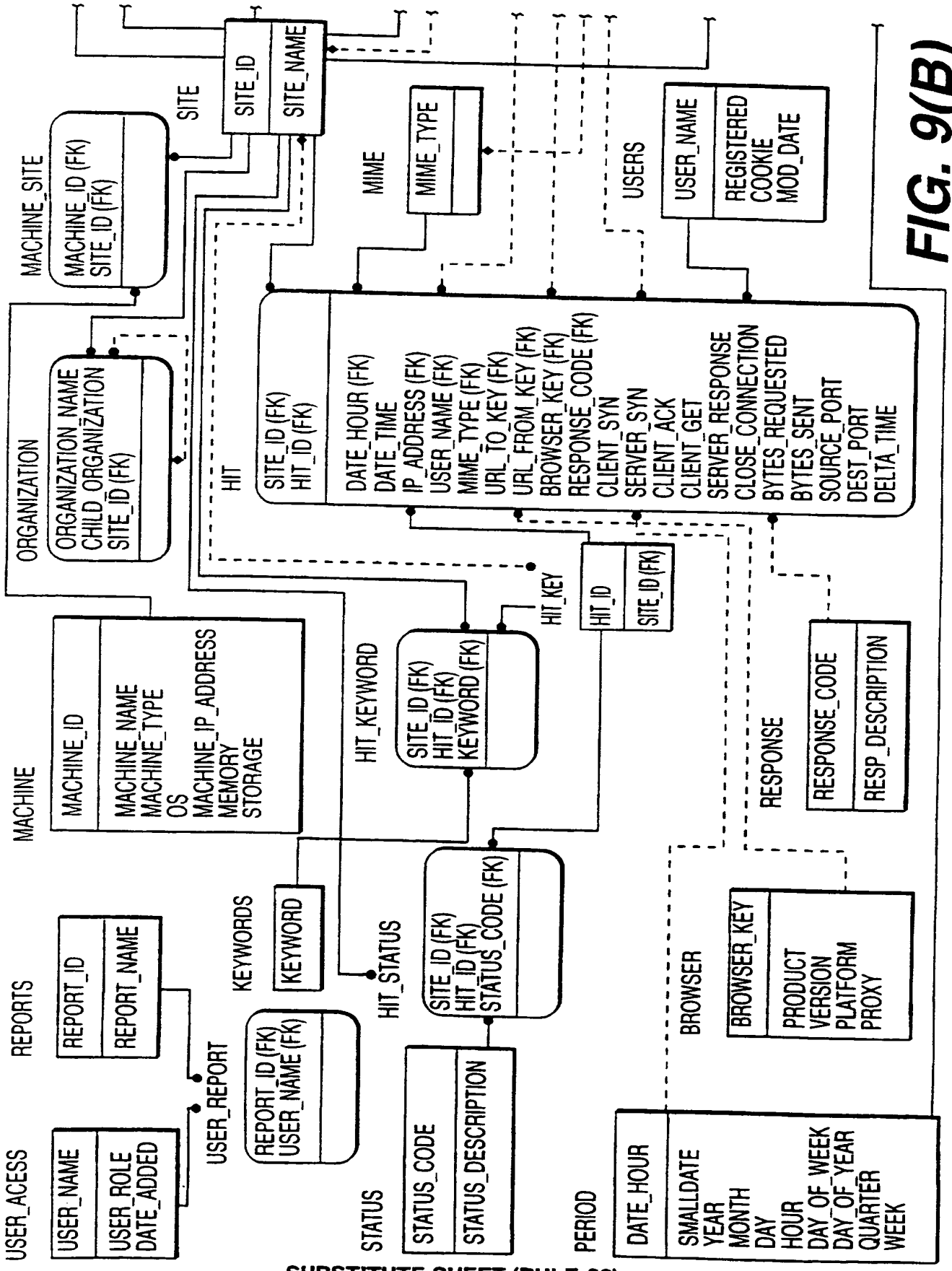

SITE
SITE_ID
SITE_NAME
MACHINE_SITE
MACHINE_ID (FK)
SITE_ID (FK)
MIME
MIME_TYPE
USERS
USER_NAME
REGISTERED
COOKIE
MOD_DATE
ORGANIZATION
ORGANIZATION_NAME
CHILD_ORGANIZATION
SITE_ID (FK)
HIT
SITE_ID (FK)
HIT_ID (FK)
DATE_HOUR (FK)
DATE_TIME
IP_ADDRESS (FK)
USER_NAME (FK)
MIME_TYPE (FK)
URL_TO_KEY (FK)
URL_FROM_KEY (FK)
BROWSER_KEY (FK)
RESPONSE_CODE (FK)
CLIENT_SYN
SERVER_SYN
CLIENT_ACK
CLIENT_GET
SERVER_RESPONSE
CLOSE_CONNECTION
BYTES_REQUESTED
BYTES_SENT
SOURCE_PORT
DEST_PORT
DELTA_TIME
HIT_KEY
HIT_ID
SITE_ID (FK)
MACHINE
MACHINE_ID
MACHINE_NAME
MACHINE_TYPE
OS
MACHINE_IP_ADDRESS
MEMORY
STORAGE
HIT_KEYWORD
SITE_ID (FK)
HIT_ID (FK)
KEYWORD (FK)
RESPONSE
RESPONSE_CODE
RESP_DESCRIPTION
BROWSER
BROWSER_KEY
PRODUCT
VERSION
PLATFORM
PROXY
KEYWORDS
KEYWORD
HIT_STATUS
SITE_ID (FK)
HIT_ID (FK)
STATUS_CODE (FK)
REPORTS
REPORT_ID
REPORT_NAME
USER_REPORT
REPORT_ID (FK)
USER_NAME (FK)
STATUS
STATUS_CODE
STATUS_DESCRIPTION
USER_ACESS
USER_NAME
USER_ROLE
DATE_ADDED
PERIOD
DATE_HOUR
SMALLDATE
YEAR
MONTH
DAY
HOUR
DAY_OF_WEEK
DAY_OF_YEAR
QUARTER
WEEK

FIG. 9(B)

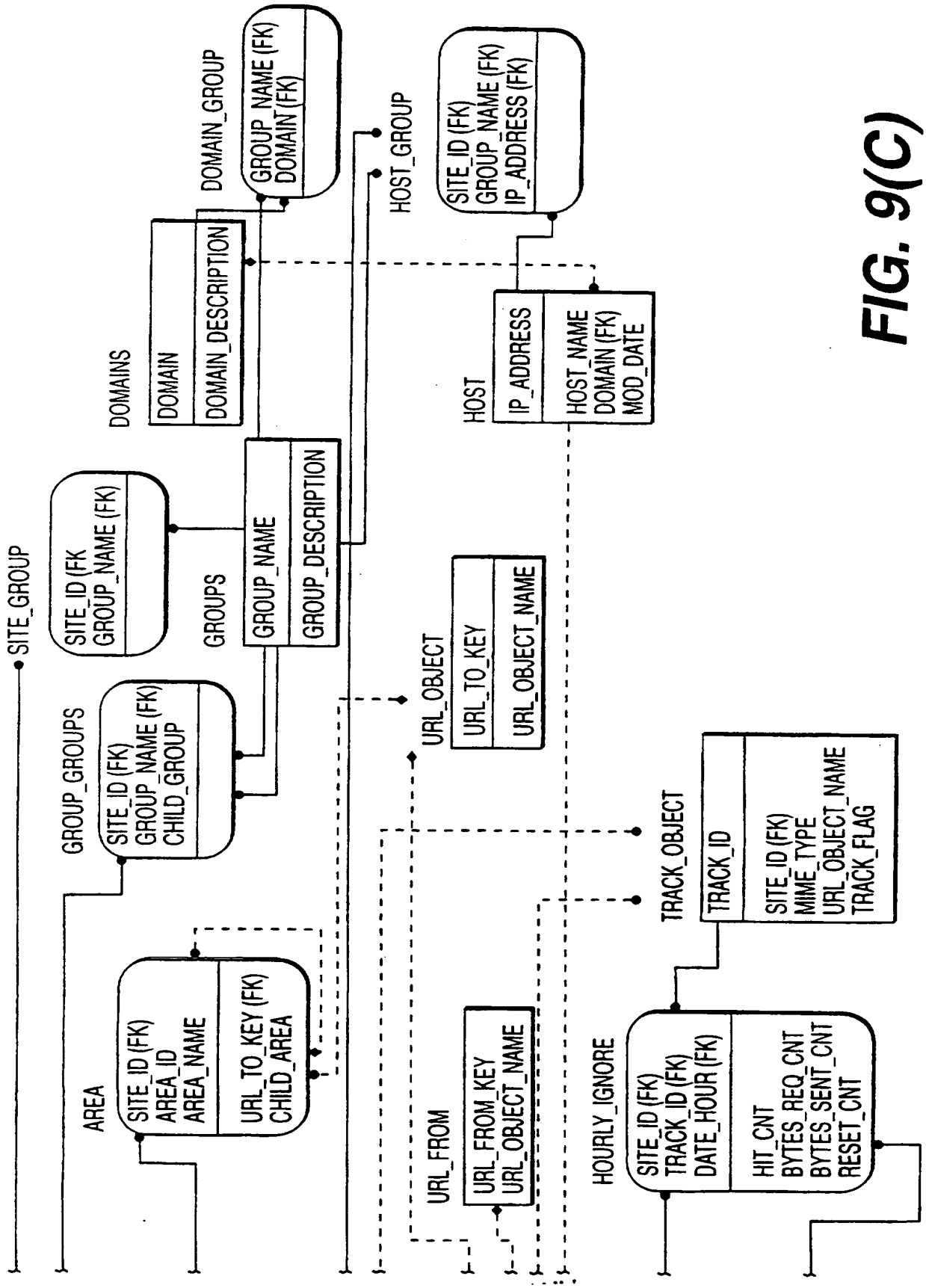

SITE_GROUP
SITE_ID (FK
GROUP_NAME (FK)
GROUPS
GROUP_NAME
GROUP_DESCRIPTION
DOMAIN_GROUP
GROUP_NAME (FK)
DOMAIN (FK)
DOMAINS
DOMAIN
DOMAIN_DESCRIPTION
HOST_GROUP
SITE_ID (FK)
GROUP_NAME (FK)
IP_ADDRESS (FK)
HOST
IP_ADDRESS
HOST_NAME
DOMAIN (FK)
MOD_DATE
GROUP_GROUPS
SITE_ID (FK)
GROUP_NAME (FK)
CHILD_GROUP
URL_OBJECT
URL_TO_KEY
URL_OBJECT_NAME
AREA
SITE_ID (FK)
AREA_ID
AREA_NAME
URL_TO_KEY (FK)
CHILD_AREA
URL_FROM
URL_FROM_KEY
URL_OBJECT_NAME
TRACK_OBJECT
TRACK_ID
SITE_ID (FK)
MIME_TYPE
URL_OBJECT_NAME
TRACK_FLAG
HOURLY_IGNORE
SITE_ID (FK)
TRACK_ID (FK)
DATE_HOUR (FK)
HIT_CNT
BYTES_REQ_CNT
BYTES_SENT_CNT
RESET_CNT

FIG. 9(C)

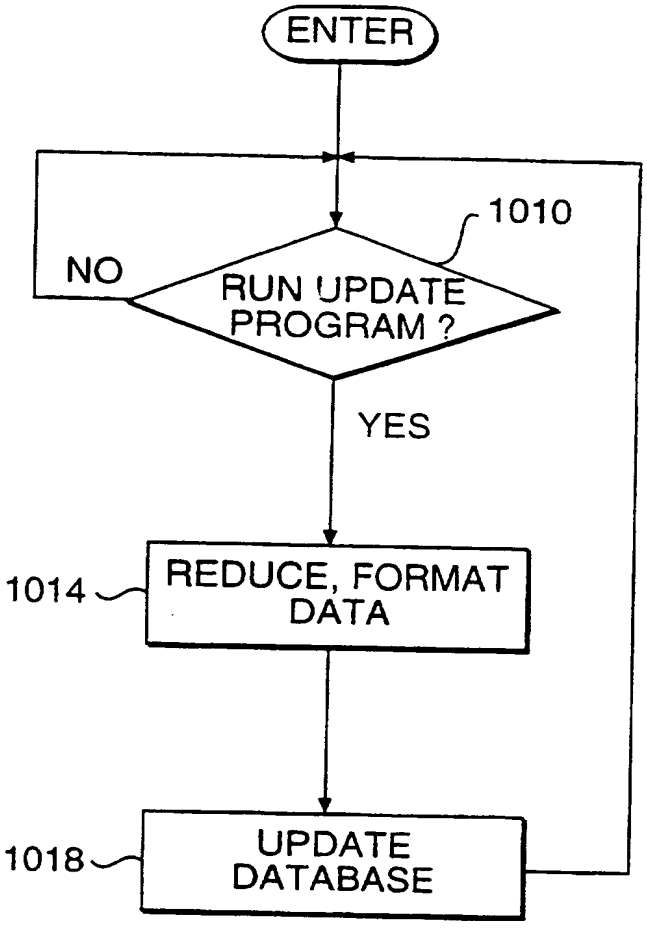

ENTER
1010
NO
RUN UPDATE PROGRAM ?
YES
1014
REDUCE, FORMAT DATA
1018
UPDATE DATABASE

FIG. 10

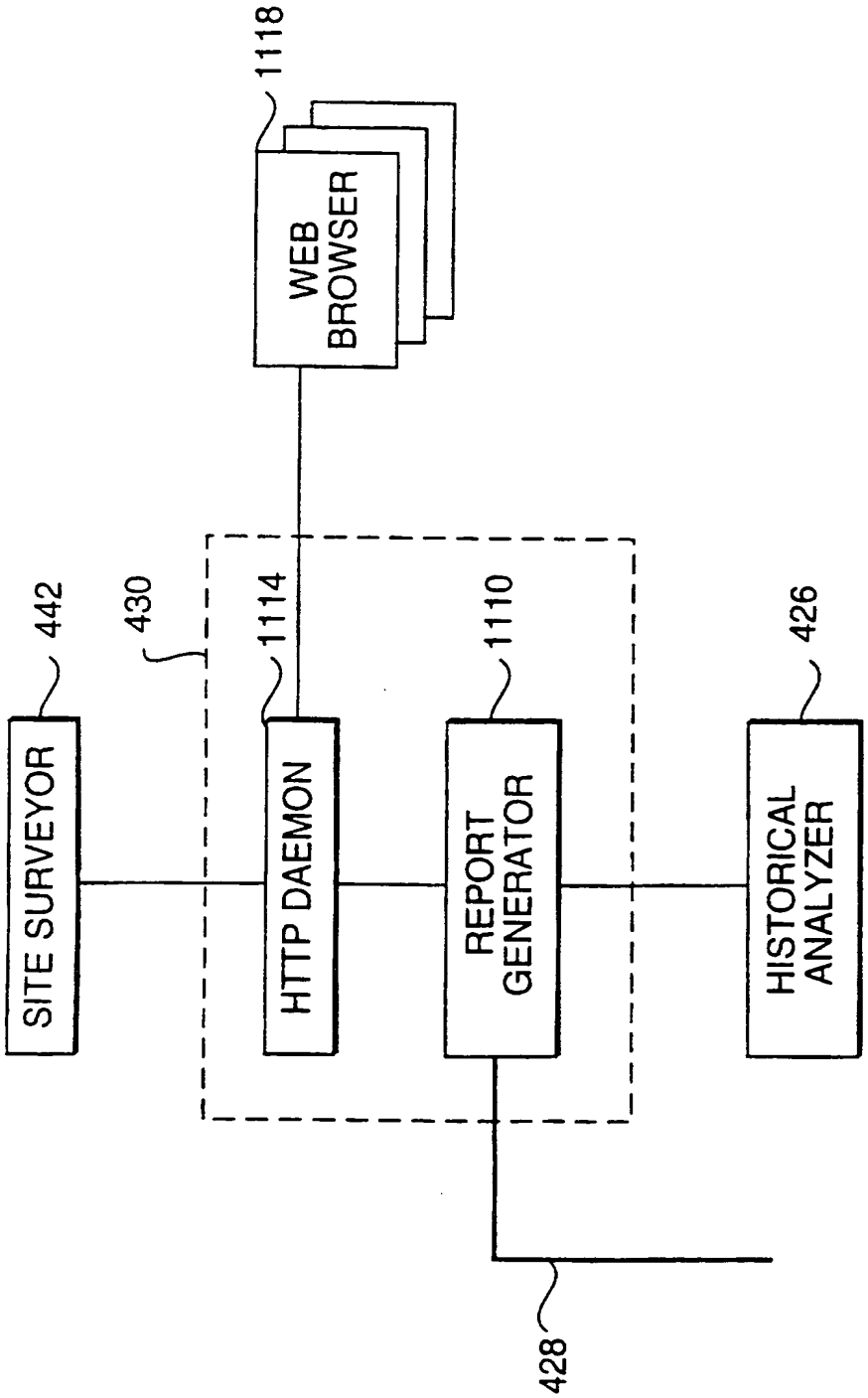


FIG. 11

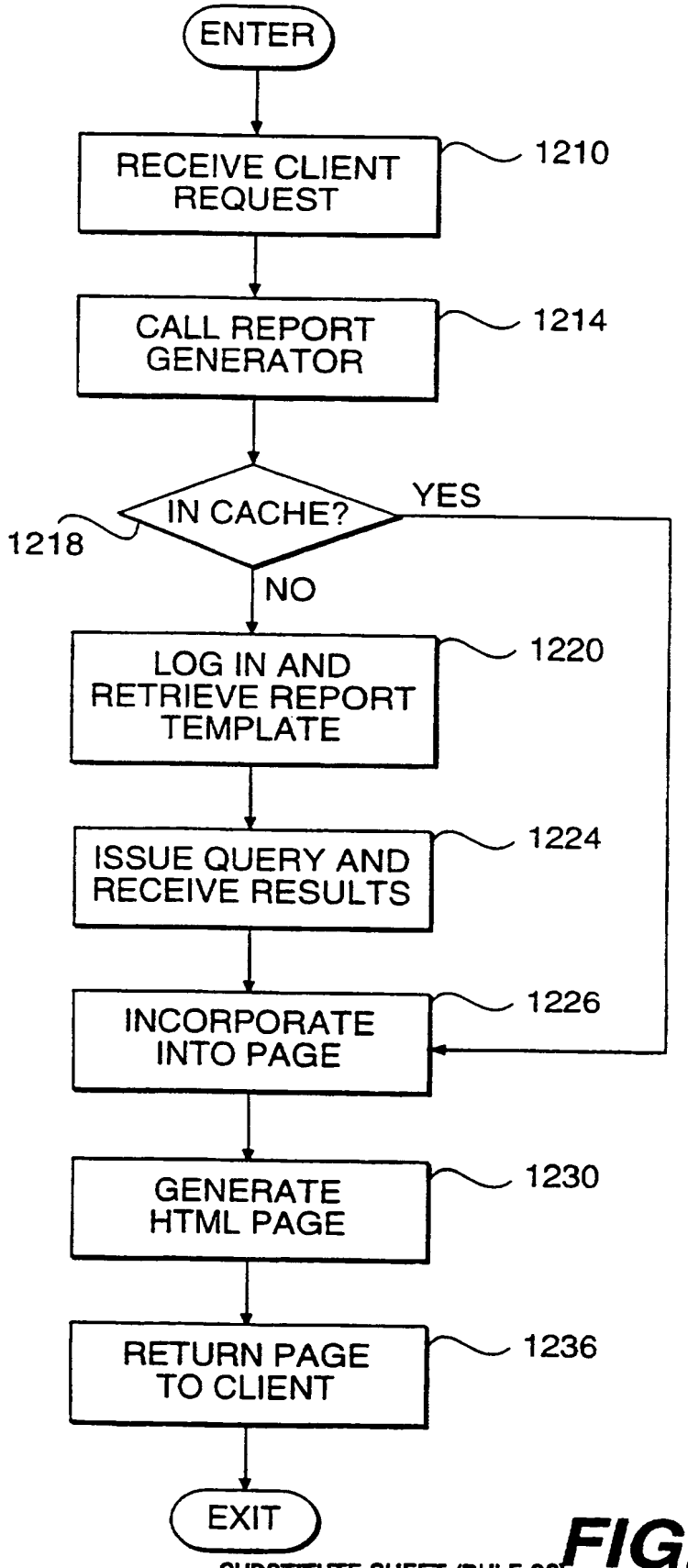


FIG. 12

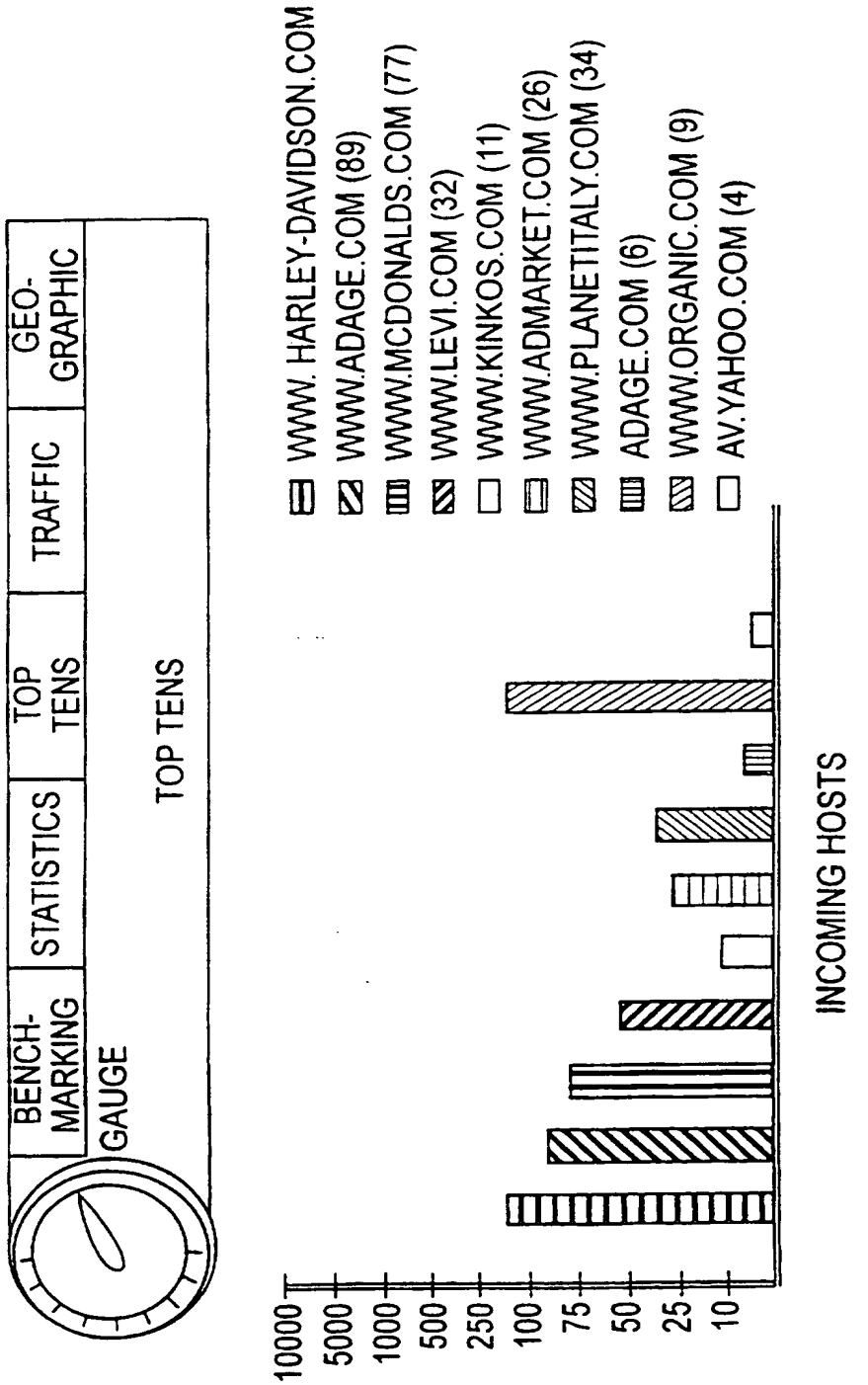

BENCH-MARKING
STATISTICS
TOP TENS
TRAFFIC
GEO-GRAPHIC
GAUGE
TOP TENS
WWW. HARLEY-DAVIDSON.COM
WWW.ADAGE.COM (89)
WWW.MCDONALDS.COM (77)
WWW.LEVI.COM (32)
WWW.KINKOS.COM (11)
WWW.ADMARKET.COM (26)
WWW.PLANETITALY.COM (34)
ADAGE.COM (6)
WWW.ORGANIC.COM (9)
AV.YAHOO.COM (4)
10000
5000
1000
500
250
100
75
50
25
10
INCOMING HOSTS

FIG. 13(A)

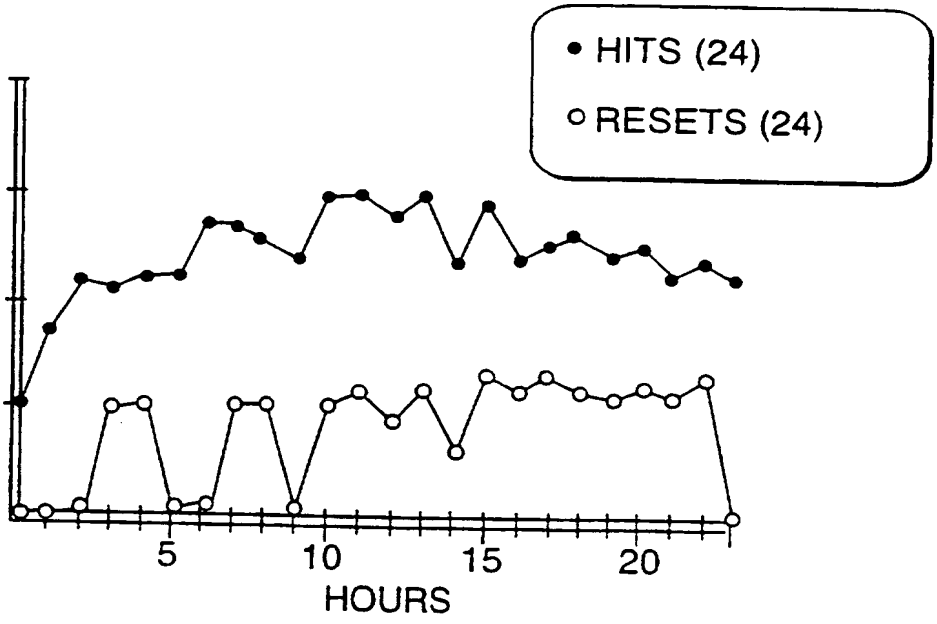


FIG. 13(B)

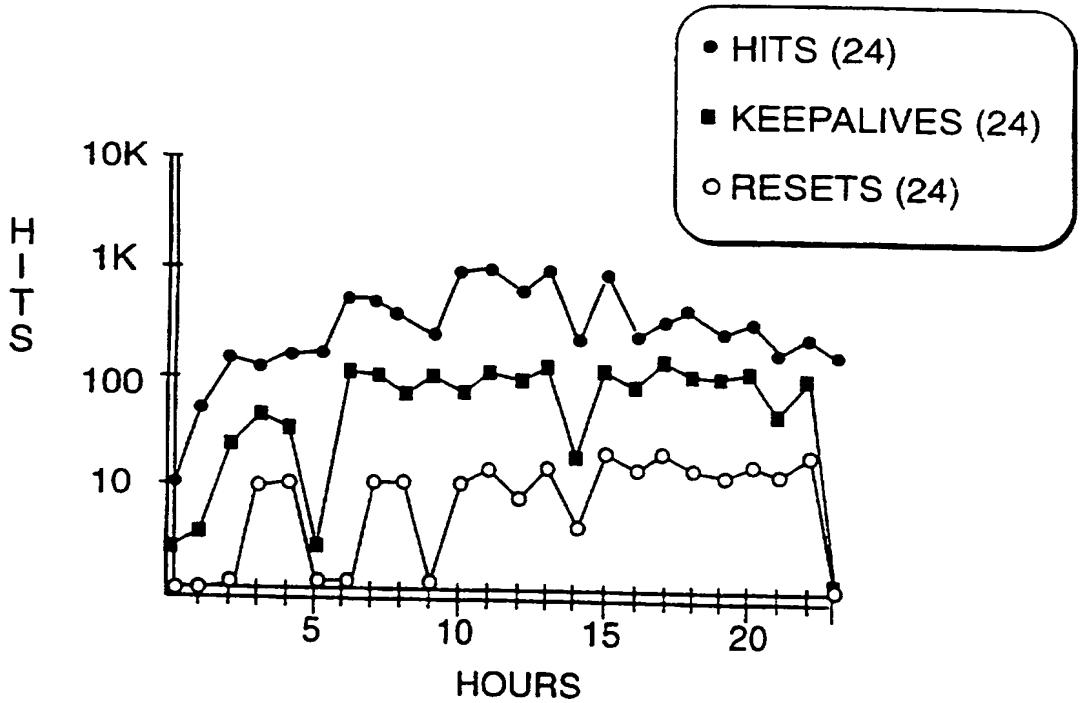


FIG. 13(C)

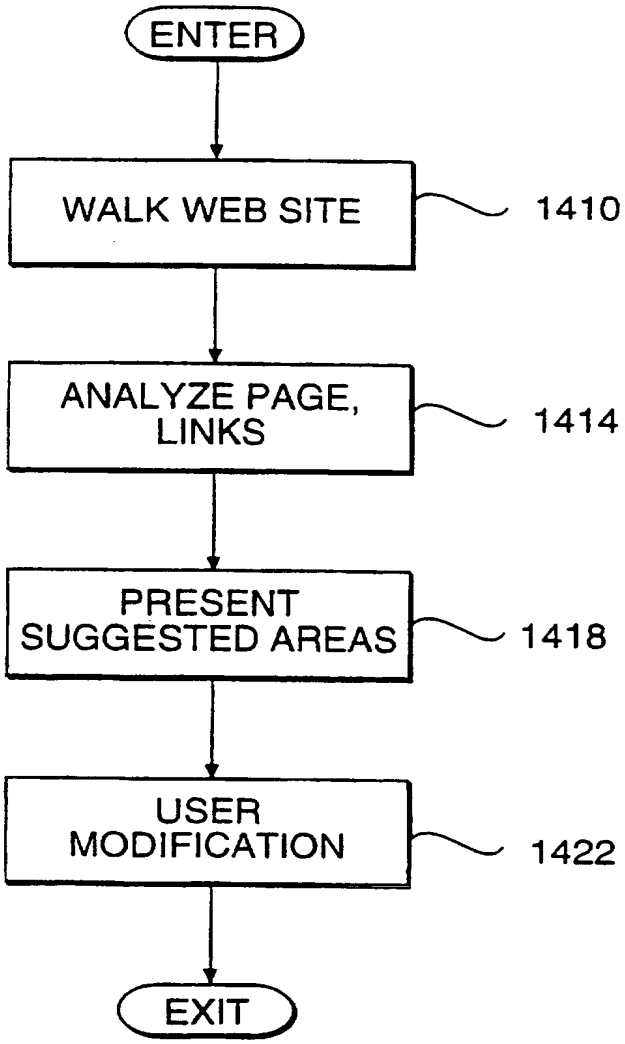

ENTER
WALK WEB SITE
1410
ANALYZE PAGE, LINKS
1414
PRESENT SUGGESTED AREAS
1418
USER MODIFICATION
1422
EXIT

FIG. 14

INTERNATIONAL SEARCH REPORT

Int. ional Application No
PCT/US 97/15837

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L29/06 H04L12/26

According to International Patent Classification(IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No
	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

1

Date of the actual completion of the international search	Date of mailing of the international search report
15 January 1998	28/01/1998

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Vaskimo, K

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inter ...nal Application No
PCT/US 97/15837

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5101402 A	31-03-92	NONE	
EP 0477448 A	01-04-92	CA 2044874 A	29-03-92
		DE 69020899 D	17-08-95
		DE 69020899 T	07-12-95
		DE 69122200 D	24-10-96
		DE 69122200 T	30-01-97
		EP 0480555 A	15-04-92
		WO 9206547 A	16-04-92
		JP 4263536 A	18-09-92
		JP 5502566 T	28-04-93
		US 5315580 A	24-05-94
		US 5450408 A	12-09-95
EP 0478175 A	01-04-92	EP 0474932 A	18-03-92
		DE 69114805 D	04-01-96
		DE 69114805 T	18-04-96
		US 5347524 A	13-09-94
US 5535193 A	09-07-96	CA 2159301 A	10-08-96
		EP 0726664 A	14-08-96
		JP 8251167 A	27-09-96
US 5572533 A	05-11-96	JP 8116334 A	07-05-96
US 5351243 A	27-09-94	NONE	
EP 0332286 A	13-09-89	JP 2010954 A	16-01-90
EP 0265106 A	27-04-88	US 4775973 A	04-10-88
		CA 1279917 A	05-02-91
		DE 3788189 D	23-12-93
		DE 3788189 T	19-05-94
		HK 7894 A	04-02-94
		JP 2012706 C	02-02-96
		JP 7036555 B	19-04-95
		JP 63107249 A	12-05-88
		SG 135893 A	31-03-94

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 97/15837

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 265 106 A (HEWLETT PACKARD CO) 27 April 1988	1-3,6,8, 12-14, 16,17, 20-23, 25,26, 29-31, 34,35, 38,40, 48-50, 58-61
A	JANDER M: "A HANDY WAY TO ANALYZE LAN STATS STAR-TEK'S HAND-HELD DEVICE FILTERS, CAPTURES, AND ANALYZES PACKETS FROM ETHERNETS AND TOKEN RINGS" DATA COMMUNICATIONS, vol. 21, no. 14, 1 October 1992, pages 45-46, XP000316517 see the whole document	1-3,6,8, 12,16, 17,20, 21,25, 26, 29-31, 34,35, 38,40, 48-50, 58-61
A	EDMUND G. MOORE: "The HP Network Advisor: A Portable Test Tool for Protocol Analysis" HEWLETT-PACKARD JOURNAL, October 1992, pages 6-10, XP002052182 see page 6, left-hand column, line 1 - page 7, right-hand column, line 6 see page 7, right-hand column, line 53 - page 8, left-hand column, line 16 see page 9	1,2,8, 12,16, 21,25, 30,34, 40,48, 50,58, 60,61

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	MATHIS M: "WINDOWED PING: AN IP LAYER PERFORMANCE DIAGNOSTIC" COMPUTER NETWORKS AND ISDN SYSTEMS, vol. 27, no. 3, 1 December 1994, pages 449-459, XP000483277 see abstract see paragraph 3.1. - paragraph 3.2. see figures 1,2	7
A		2,16,25, 34,48, 58,61

Y	US 5 351 243 A (KALKUNTE RAMESH S ET AL) 27 September 1994 see column 1, line 12 - column 2, line 23 see column 3, line 3 - column 5, line 2 see column 10, line 3 - column 11, line 9 see column 12, line 38 - column 13, line 37 see column 25, line 16 - column 26, line 34 see column 30, line 53 - column 31, line 42	10
A		2,3,6, 16,17, 20,25, 26,29, 31,34, 35,38, 48,49, 58,59,61

Y	BURDICK M J: "CONTINUOUS MONITORING OF REMOTE NETWORKS: THE RMON MIB" HEWLETT-PACKARD JOURNAL, vol. 44, no. 2, 1 April 1993, pages 82-89, XP000360816 see abstract see page 85, right-hand column, line 8 - page 88, right-hand column, line 4	63
A		2,16,25, 34,48, 58,61,64

A	EP 0 332 286 A (HEWLETT PACKARD CO) 13 September 1989 see abstract see page 2, line 4 - line 44 see page 3, line 31 - page 5, line 39	1,2,8, 12,16, 21,25, 30,34, 39,40, 48,50, 58,60,61

	-/--	

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passagee	Relevant to claim No.
P,Y	US 5 572 533 A (SUNADA KAZUHIRO ET AL) 5 November 1996 see column 2, line 8 - column 4, line 67 see column 5, line 43 - column 8, line 36	4,5,10, 11,18,19
A		1,2, 8-10,12, 14-16, 21,23, 30-34, 36,37, 40,50, 60-63

Y	PATENT ABSTRACTS OF JAPAN vol. 096, no. 009, 30 September 1996 & JP 08 116334 A (FUJITSU LTD), 7 May 1996, see abstract	4,5,10, 11,18,19
A		1,2, 8-10,12, 14-16, 21,23, 30-34, 36,37, 40,50, 60-63

Y	SUNIL BHAT: "THE NETWORK ADVISOR ANALYSIS AND REAL-TIME ENVIRONMENT" HEWLETT-PACKARD JOURNAL, vol. 43, no. 5, 1 October 1992, pages 29-33, XP000349771 see page 29, left-hand column, line 1 - right-hand column, line 33 see page 31, right-hand column, line 13 - line 38	13,14, 22,23
A		2,3,6, 12,16, 17,20, 21,25, 26,29, 31,34, 35,38, 48,49, 58,59,61

	-/--	

1

INTERNATIONAL SEARCH REPORT

Inte ional Application No
PCT/US 97/15837

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	EP 0 478 175 A (HEWLETT PACKARD CO) 1 April 1992 see column 2, line 13 - column 4, line 46 see column 6, line 24 - column 10, line 48	7,13,14, 22,23, 30-32, 41,44, 50,51, 54,68,69
A		1,2,8, 12,16, 21,25, 33-35, 39-44, 48, 51-54, 58,60,61

Y	US 5 535 193 A (ZHANG JING ET AL) 9 July 1996 see column 1, line 13 - line 62 see column 3, line 62 - column 5, line 60	4,5,18, 19,27, 28,63,64
A		2,3,6,7, 16,20, 25,26, 29,31, 34,35, 38,48, 49,58, 59,61

Y	MOGUL J C: "NETWORK LOCALITY AT THE SCALE OF PROCESSES" PROCEEDINGS OF THE CONFERENCE ON COMMUNICATIONS ARCHITECTURES AND PROTOCOLS (SIGCOMM), ZURICH, SEPT. 3 - 6, 1991, vol. 21, 3 September 1991, ASSOCIATION FOR COMPUTING MACHINERY, pages 273-284, XP000301959 see paragraph 1. - paragraph 3.4. see paragraph 5. see tables 5-1	7,41,50, 51,55-59
A		2,14,16, 23,25, 34,42, 43,48, 52,54, 58,61

	-/--	

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Int. ional Application No
PCT/US 97/15837

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 101 402 A (CHIU DAH-MING ET AL) 31 March 1992	1-3,6,8, 9,12, 15-17, 20,21, 24-26, 29,40, 45-49, 60-62, 64-66
	see column 3, line 4 - column 12, line 6	
Y		4,5,7, 10,11, 13,14, 18,19, 22,23
Y		30-32, 34,35, 37,38, 41,44, 50,51, 54-59, 63,64, 67-69
A		22,23, 27-29, 33,36, 39,43, 46,47, 52,53, 56,57,67

Y	EP 0 477 448 A (HEWLETT PACKARD CO) 1 April 1992	4,5,7, 18,19, 27,28, 30-32, 34,35, 37,38, 41,67
	see the whole document	
A		2,16,25, 33,39, 42,43, 48,58,61

	-/--	

Form PCT/ISA/210 (continuation of second sheet) (July 1992)